Ana Catarina Gralha de Almeida

Master of Science

Quality Evaluation of Requirements Models: The Case of Goal Models and Scenarios

Thesis submitted in partial fulfillment
of the requirements for the degree of

Doctor of Philosophy in
Computer Science

Adviser: Miguel Carlos Pacheco Afonso Goulão,
Assistant Professor, Universidade Nova de Lisboa

Co-adviser: João Baptista da Silva Araujo Junior,
Assistant Professor, Universidade Nova de Lisboa

Examination Committee

Chair:	José Augusto Legatheaux Martins
Rapporteurs:	John Mylopoulos
	João Carlos Pascoal Faria
Members:	Xavier Franch
	Ana Maria Diniz Moreira
	Miguel Carlos Pacheco Afonso Goulão

FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

September, 2019

Quality Evaluation of Requirements Models: The Case of Goal Models and Scenarios

This document was created using the (pdf)LaTeX processor, based on the "novathesis" template[1], developed at the Dep. Informática of FCT-NOVA [2].
[1] https://github.com/joaomlourenco/novathesis [2] http://www.di.fct.unl.pt

Em memória de Maria Madalena Ribeiro Lopes Gralha e
Joaquim de Oliveira Gralha.

Acknowledgements

Obtaining a PhD degree is a demanding journey, but full of people that have directly or indirectly helped me along the way. First and foremost, none of this would have been possible without the support, guidance, and friendship of my advisers, Professors Miguel Goulão and João Araújo. It has been a true pleasure to work and grow with you during all these years. You have encouraged me to pursue a PhD degree, and never doubted me or my abilities. It was a real privilege to work under your supervision. Thank you for all the suggestions, knowledge and insights, and also for the freedom to explore new ideas. *Obrigada!* To Professors Ana Moreira and Xavier Franch, for the interesting discussions and useful observations. A special thank you to Professor Daniela Damian, for the amazing opportunity to work in the University of Victoria, and with several software startups. For all the advice, endless support, and for making me feel at home in a foreign land. I learned a lot while in Canada, and I know I am a better researcher for it. *Mulțumesc!* To Professor Awais Rashid, for welcoming me in the University of Bristol, and for the incredible opportunity to work with the Bristol Cyber Security group. It opened my mind to new fields of research. I am also grateful to the institutions that contributed to the success of this work. To the Departamento de Informática from Universidade NOVA de Lisboa, NOVA LINCS (UID/CEC/04516/2013 and UID/CEC/04516/2019), Fundação para a Ciência e Tecnologia (FCT-MCTES SFRH/BD/108492/2015), University of Victoria, and University of Bristol, for financial support. A word of appreciation to the participants in the quasi-experiments, and to the companies that opened their doors for Science. I thank my colleagues of the P3/12 lab, for the moments of relaxation and for always being available to exchange thoughts. To João Cambeiro, Joana Pereira, Cristiano de Faveri, Denise Bombonatti, Lyrene Silva, Mafalda Santos, Rita Pereira, and Mariana Leite. To my friends, that always supported me. For all the moments of fun and laughter, and for helping me prove that we can work on a PhD and still have a vivid social life. To Andy Gonçalves, Diogo Cordeiro and Gabriel Marcondes, for 10 years of friendship. To Joana Gato, for these 25 years. You are proof that friendships can endure a separation of over 2000 kilometres. I could write an entire page to João Silva, but I will only write a few phrases. Thank you for all the encouragement and patience, and for always being there for me. For calming me down in the most stressful times, and for truly and unconditionally believing in my success. Last but not least, I would like to thank my mother, for the education, and for all the incentives provided over the years.

Abstract

Context: Requirements Engineering approaches provide expressive model techniques for requirements elicitation and analysis. Yet, these approaches struggle to manage the quality of their models, causing difficulties in understanding requirements, and increase development costs. The models' quality should be a permanent concern. **Objectives:** We propose a mixed-method process for the quantitative evaluation of the quality of requirements models and their modelling activities. We applied the process to goal-oriented (*i** 1.0 and iStar 2.0) and scenario-based (ARNE and ALCO use case templates) models, to evaluate their usability in terms of appropriateness recognisability and learnability. We defined (bio)metrics about the models and the way stakeholders interact with them, with the GQM approach. **Methods:** The (bio)metrics were evaluated through a family of 16 quasi-experiments with a total of 660 participants. They performed creation, modification, understanding, and review tasks on the models. We measured their accuracy, speed, and ease, using metrics of task success, time, and effort, collected with eye-tracking, electroencephalography and electro-dermal activity, and participants' opinion, through NASA-TLX. We characterised the participants with GenderMag, a method for evaluating usability with a focus on gender-inclusiveness. **Results:** For *i**, participants had better performance and lower effort when using iStar 2.0, and produced models with lower accidental complexity. For use cases, participants had better performance and lower effort when using ALCO. Participants using a textual representation of requirements had higher performance and lower effort. The results were better for ALCO, followed by ARNE, iStar 2.0, and *i** 1.0. Participants with a comprehensive information processing and a conservative attitude towards risk (characteristics that are frequently seen in females) took longer to start the tasks but had a higher accuracy. The visual and mental effort was also higher for these participants. **Conclusions:** A mixed-method process, with (bio)metric measurements, can provide reliable quantitative information about the success and effort of a stakeholder while working on different requirements models' tasks.

Keywords: requirements models, quality, usability, *i**, use cases, (bio)metrics, gender

Resumo

Contexto: As abordagens de Engenharia de Requisitos oferecem técnicas de modelação expressivas, que ajudam na elicitação e análise de requisitos. Porém, estas abordagens apresentam problemas na qualidade dos seus modelos, contribuindo para dificuldades na compreensão e aumento dos custos de desenvolvimento. A qualidade dos modelos deve ser uma preocupação constante. **Objetivos:** Propõe-se um processo com um método misto para avaliar a qualidade de modelos de requisitos. O processo foi aplicado a modelos orientados a objetivos (*i** 1.0 e iStar 2.0), e baseados em cenários (casos de uso ARNE e ALCO), para avaliar a sua usabilidade em termos de compreensão e aprendizagem. Foram definidas (bio)métricas sobre os modelos e como *stakeholders* interagem com eles, usando GQM. **Métodos:** As (bio)métricas foram avaliadas com uma família de 16 quasi-experiências com um total de 660 participantes, que realizaram tarefas de criação, modificação, compreensão e revisão nos modelos. Foram medidas a sua exatidão, velocidade e facilidade, com métricas de sucesso, tempo e esforço, recolhidas com *eye-tracking*, eletroencefalografia e atividade eletro-dérmica, e a opinião dos participantes, através de NASA-TLX. Os participantes foram caracterizados com GenderMag, para avaliação da usabilidade com foco em inclusão de género. **Resultados**: No *i**, os participantes tiveram melhor desempenho e menor esforço quando usaram iStar 2.0, e produziram modelos com menor complexidade acidental. Nos casos de uso, tiverem melhor desempenho e menor esforço quando usaram ALCO. Os participantes que usaram uma representação de requisitos textual tiveram maior precisão e menor esforço, com resultados superiores para ALCO, seguido de ARNE, iStar 2.0, e *i** 1.0. Participantes com um processamento de informação amplo e aversos ao risco (características frequentes em mulheres) demoraram mais a começar, mas tiveram maior exatidão. O seu esforço visual e mental também foi maior. **Conclusões:** Um processo com um método misto, recorrendo a (bio)métricas, oferece informação quantitativa fiável relativa ao sucesso e esforço de um *stakeholder* quando realiza tarefas em modelos de requisitos.

Palavras-chave: modelos de requisitos, qualidade, usabilidade, *i**, casos de uso, (bio) métricas, género

Contents

List of Figures

List of Tables

Listings

Acronyms

ALCO Use case template proposed by **Al**istair **Co**ckburn [42]

ARNE Use case template proposed by **Ar**low and **Ne**ustadt [5]

EDA Electro-dermal Activity

EEG Electroencephalography

GenderMag Gender Inclusiveness Magnifier

GQM Goal-Question-Metric

NASA-TLX NASA-Task Load Index

NN50 Number of pairs of successive beat-to-beat intervals that differ more than 50ms

RE Requirements Engineering

RMSSD Root mean square of successive differences of two heart beats

Chapter 1

Introduction

The present dissertation has it roots in Requirements Engineering (RE) and Empirical Software Engineering. RE success depends on, among several other factors, the quality of the communication between requirements engineers and other stakeholders. Indeed, communication flaws are among the most frequently reported RE problems that may lead to project failure. One of the key elements of an effective communication is the usability of the requirements models used. However, RE approaches are still encountering challenges when it comes to managing the quality of the requirements models. In this dissertation, we are using Empirical Software Engineering techniques to identify the strengths and shortcomings in the usability of requirements models. In this introductory Chapter, we present the context and motivation for the research work undertaken, as well as the main goals and contributions. We conclude with the outline of this document.

1.1 Context and Motivation

RE approaches, following paradigms such as goal-oriented [129] or scenario-based [211], provide expressive model techniques (both diagrammatic and textual) for requirements elicitation and analysis. These requirements models are often used for communication with different types of stakeholders. For this communication to be effective, both requirements engineers and other stakeholders need to have a common understanding of the requirements models [32]. However, as a prevailing challenge, RE approaches are still struggling when it comes to managing the quality of their models. Quality problems can cause difficulties in the management and understanding of those models and the requirements they represent, leading to increased development costs [52, 211]. These difficulties in understanding the model can introduce validation errors: a stakeholder

may incorrectly understand a given model (due to its accidental complexity [23], for example) and accept a specification that does not meet his needs. Other problems in quality, such as incomplete or unnecessarily complex specifications, may jeopardise the correct implementation of the software-intensive system.

Requirements elicitation and analysis is a particularly critical activity on the software development process, as errors at this stage inevitably lead to later problems in the system design and implementation [170]. Studies performed at several companies have measured and assigned costs to errors occurring at various stages of a software system lifecycle. Davis [52] summarised a number of these studies. Although they were run independently, all the studies reached relatively the same conclusion: if a unit cost of 1 (one) is assigned to the effort required to detect and repair an error introduced during the coding stage, then the cost to detect and repair an error, in the coding stage, that was introduced during the requirements stage, is between 5 (five) to 10 (ten) times more. Moreover, the cost to detect and repair that error during maintenance is 20 (twenty) times more. Thus, the cost of repairing an error made in the requirements elicitation stage increases along the next stages of a software project [52]. Altogether, this means that a 200:1 cost savings results from finding errors in the requirements stage *versus* finding them in the maintenance stage. More recently, Chari and Agrawal [38] analysed a sample of 49 software projects following the Waterfall methodology, from organisations with CMMI level 5 [41]. They concluded that the resolution of change requests due to new requirements increases defects injected as well as effort. Furthermore, the resolution of change requests due to incorrect requirements increases the number of new requirements, as well as the number of defects injected. For these reasons, it is imperative that requirements-related problems are detected and solved as early as possible.

One might think that these concerns are not applicable to the fast-paced world of startups, since they typically follow agile practices [65], where errors and small problems can be quickly fixed in the next release (which can be deployed in the same day) [72]. However, startups intentionally introduce technical debt in order to reduce development time. This technical debt becomes more severe as the product grows, and introduces future development risks [16]. To the best of our knowledge, there is no exact number for the cost of repairing this technical debt. However, research has shown that it negatively influences the future growth of startup companies [48]. Furthermore, and while the initial requirements may have been created informally, the informal approach does not often scale well as the company grows [72, 83]. Moreover, non-functional requirements are often not addressed by these companies, which causes further problems when growing the product or service [72]. Many startups fail to adapt their requirements practices and shut down within their first two years [167]. In that sense, requirements elicitation problems may have a deadly cost for startups. Hence, these issues also affect companies that do not follow a well-defined RE process.

Only by understanding which are and where the quality problems that affect those models reside, is it possible to identify effective opportunities for their improvement. In

that sense, quality attributes of the models, such as usability, should be measured and monitored during and after the requirements modelling activity, that is, when the model is being built, and when it is being used. Evaluating usability (in particular, in terms of appropriateness recognisability and learnability) while the models are being built can give us insights about how the model is created, and what the actual effort required for both its creation and modification is. On the other hand, post-mortem analysis can support an evidence-based understanding on how the modelling language constructs are used, in practice. Furthermore, it can provide useful information about the actual relationship between different types of stakeholders and the model, in terms of their ability and effort to understand and review it, and by providing data on which specific parts of the models are more problematic.

With a quantitative assessment of the requirements models' quality, it is then possible to promote adjustments and changes in the development process, to mitigate the causes of problems that significantly affect the production of a software-intensive system. In the end, by identifying quality problems, it is possible to characterise and analyse them to look for patterns of a wrong usage or wrong understanding of the modelling language. This type of information can also provide useful insights for the evolution process of the modelling approaches themselves.

1.2 Goals and Research Questions

In this dissertation, we propose a **mixed-method process for the quantitative evaluation of the quality of requirements models**. This goal is based on the previously described notion that we first need to understand which are and where the quality problems that affect requirements models reside, to then identify effective opportunities for their improvement. However, using a single type of measurement in a quantitative evaluation introduces a risk. If there is a measurement bias, then the results will be misleading. By using different types of measures, they can be cross-checked against each other, producing stronger and more solid results [245]. As such, the general research question of this dissertation is:

How can we leverage a mixed-method process to characterise the quality of requirements models and the way stakeholders interact with them?

In particular, our research on using a mixed-method process can be divided into a more specific research question, with the corresponding hypotheses:

RQ How can (bio)metric measurements be used to understand whether tasks such as creating, modifying, understanding and reviewing requirements models are difficult or easy to perform by a given stakeholder?

H_0 (Bio)metric measurements **do not** provide reliable quantitative information about the success and effort a stakeholder experiences while working on different requirements models' tasks.

H_1 (Bio)metric measurements provide reliable quantitative information about the success and effort a stakeholder experiences while working on different requirements models' tasks.

To answer this question and explore the corresponding hypotheses, we investigate the particular case of **goal-oriented** and **scenario-based** models. We selected ***i** 1.0** [247, 248] and **iStar 2.0** [49], combining goal and agent-oriented approaches. For the scenario-based approach, we selected 2 (two) different use cases templates: the one proposed by **Ar**low and **Ne**ustadt [5], which we named **ARNE use case template**; and the one proposed by **Al**istair **Co**ckburn [42], which we named **ALCO use case template**.

The *i** framework is widely used in academia [97], with working groups in over 20 countries [243] and several editions of international workshops [241]. Furthermore, the appearing of iStar 2.0 introduced the need for studies about its ease of use, adequacy for teaching, expressiveness, graphical notation, among others [49]. Use cases, in turn, are commonly used by practitioners in several software companies, especially in their free textual form and text with constraints [151]. The ones we selected for evaluation are widely known and accepted [226]. ARNE is simpler and ALCO is more complete, which serves the purpose of contrasting a simpler with a more detailed use case specification template. Furthermore, we want to contrast the efficiency of textual and diagrammatic representations of requirements, with participants familiar and unfamiliar with these representations, in particular to understand if there is a requirements representation better suited for a particular type of stakeholders. At the methodological level, we are interested in understanding if a mixed-method process can be applied to both textual and diagrammatic representations of requirements.

Our focus is on the **usability** of these requirements models, in terms of **appropriateness recognisability** and **learnability** [107]. This is aimed to tackle requirements reading and writing, that is, the understandability of previously defined requirements *(appropriateness recognisability)*, and the ability to properly describe them *(learnability)*.

To this end, we propose a step-by-step guide on how to perform a quantitative evaluation of the quality of requirements models, by conducting (quasi-)experiments involving human subjects, and with the usage of (bio)metrics. We define metrics about *i** and incorporate them into a modelling and measurement tool, so that they can be automatically collected. We then perform a family of 16 quasi-experiments with different types of participants. The usability of the models is measured by a combination of these metrics and by collecting biometric data from stakeholders, by using eye-tracking devices, electroencephalography (EEG), and electro-dermal activity (EDA) scanners, while stakeholders are creating, modifying, understanding and reviewing these models. Furthermore, we also collect information about the success and speed, as well as subjective opinion of

stakeholders about the usage of these models, through the NASA-TLX questionnaire [94], which measures perceived effort while working on tasks. Finally, we characterise the participants according to GenderMag [27], a method for evaluating usability with a focus on gender-inclusiveness, in order to evaluate the impact of cognitive differences related to gender on the appropriateness recognisability and learnability of the analysed requirements models. The combination of all these techniques gives us a multi-perspective of the models, the problems they may have, and the way stakeholders interact with them.

1.3 Main Contributions

The results of this dissertation contribute to software development in general, and RE in particular, with a better understanding of requirements models, the problems they may have, and the way different stakeholders interact with them. They also contribute to stakeholders engagement and empowerment. In particular, the main contributions of this dissertation are:

1. A generic mixed-method process, named QualitEva, for the quality evaluation of requirements models, which can be applied to various requirements models and quality characteristics.

2. A set of (bio)metrics for the evaluation of requirements models.

3. A quantitative evaluation providing empirical evidence on the usability of *i**, iStar 2.0, ARNE use case template, and ALCO use case template, in terms of appropriateness recognisability and learnability, by using a combination of (bio)metrics, in the tasks of creating, modifying, understanding and reviewing those models.

4. A quantitative evaluation providing empirical evidence on the differences between *i** and use cases, in the tasks of creating, modifying, understanding and reviewing those models.

5. Two online and installation-free modelling and measurement tools, which automatically collect metrics about *i** 1.0 and iStar 2.0 models' complexity and completeness.

6. An online replication package with all the materials used in the quasi-experiments reported in this dissertation, for facilitating independent replications.

In Table 1.1 we present a list of publications and projects we were involved in. The highlighted publications indicate first-authored papers and those directly related with this dissertation. The other authored publications inspired and supported the work presented in this dissertation, but are not core of this research work.

Table 1.1: List of publications and projects.

Paper	The Evolution of Requirements Practices in Software Startups. **Catarina Gralha**, Daniela Damian, Anthony Wasserman, Miguel Goulão, João Araújo. *Proceedings of the 40th International Conference on Software Engineering (ICSE 2018)* [83].
Summary	We wanted to understand, among others, if the choice of evaluation the quality of use cases was valid for both mature and startup companies. We used the Straussian Grounded Theory [46] approach to study the evolution of requirements practices of 16 (sixteen) software startups as they grow. Our theory describes the evolution of practice along 6 dimensions that emerged as relevant to their requirements activities: requirements artefacts, knowledge management, requirements-related roles, planning, technical debt and product quality. The theory also explains the turning points that drove the evolution along these dimensions.
Paper	Analysing gender differences in building social goal models: a quasi-experiment. **Catarina Gralha**, Miguel Goulão, João Araújo. *Proceedings of the 27th International Requirements Engineering Conference (RE 2019)* [84]. **(Candidate for Best Paper Award)**
Summary	We performed a quasi-experiment to evaluate the impact of different levels of GenderMag facets on creating and modifying iStar 2.0 models. We characterised 100 participants according to each GenderMag facet, and measured their accuracy, speed, and ease, using metrics of task success, time, and effort, collected with eye-tracking, EEG and EDA sensors, and participants' feedback. Although participants with facet levels frequently seen in women had lower perceived performance and speed, their accuracy was higher. We also observed some statistically significant differences in visual effort, mental effort, and stress.
Paper	Usability of requirements techniques: a systematic literature review. Denise Bombonatti, **Catarina Gralha**, Ana Moreira, João Araújo, Miguel Goulão. *Proceedings of the 31st Annual ACM Symposium on Applied Computing (SAC 2016)* [19].
Summary	To complement the technical background, we performed a systematic literature review on the usability of requirements techniques, answering the following research question: *How is the usability of requirements engineering techniques and tools addressed?* We systematically reviewed articles published in the Requirements Engineering Journal, one of the main sources for mature work in RE, and selected 19 papers. We observed that there is relatively little evidence concerning the usability of the RE approaches, denoting this has not been a top priority concern in the past. That said, we found a variety of approaches that went through some form of usability assessment, so it is fair to say the RE community is increasingly concerned about making its approaches usable for diverse stakeholders.
Paper	What is the impact of bad layout in the understandability of social goal models? Mafalda Santos, **Catarina Gralha**, Miguel Goulão, João Araújo. *Proceedings of the 24th International Requirements Engineering Conference (RE 2016)* [185].
Summary	We performed an initial quasi-experiment using eye-tracking to evaluate the effect of the layout guidelines on the i^* 1.0 novice stakeholders' ability to understand and review those models. Participants were more successful in understanding than in reviewing tasks. However, we found no statistically significant difference in the success, time taken, or perceived complexity, between tasks conducted with models with a bad layout and models with a good layout.
Paper	On the Impact of Semantic Transparency on Understanding and Reviewing Social Goal Models. Mafalda Santos, **Catarina Gralha**, Miguel Goulão, João Araújo, Ana Moreira. *Proceedings of the 26th IEEE International Requirements Engineering Conference (RE 2018)* [187]

continue on next page...

Table 1.1: *...continued from previous page*

Summary	We evaluated the impact of semantic transparency on understanding and reviewing *i** models, in the presence of a language key. We compared the standard *i** 1.0 concrete syntax with an alternative that has an increased semantic transparency. We asked 57 novice participants to perform understanding and reviewing tasks on *i** models, and measured their accuracy, speed and ease, using metrics of task success, time and effort, collected with eye-tracking and participants' feedback. We found no evidence of improved accuracy or speed attributable to the alternative concrete syntax. Although participants' perceived ease was similar, they devoted significantly less visual effort to the model and the provided language key, when using the alternative concrete syntax.
Paper	Increasing the Semantic Transparency of the KAOS Goal Model Concrete Syntax. Mafalda Santos, **Catarina Gralha**, Miguel Goulão, João Araújo. *Proceedings of the 37th International Conference on Conceptual Modeling (ER 2018)* [186]
Summary	We performed a series of related empirical experiments that include the proposal of alternative concrete syntaxes for KAOS by leveraging design contributions from novices and their evaluation with respect to semantic transparency, in contrast with the standard KAOS goal model concrete syntax. We also proposed an alternative concrete syntax for KAOS that increases its semantic transparency leading to a significantly higher correct symbol identification by novices.
Paper	Exploring Views for Goal-oriented Requirements Comprehension. Lyrene Silva, Ana Moreira, João Araújo, **Catarina Gralha**, Miguel Goulão, Vasco Amaral. *Proceedings of the 35th International Conference on Conceptual Modeling (ER 2016)* [206]
Summary	Using information visualisation techniques, we proposed three kinds of visualisations for requirements models, while providing a gradual model overview: big picture, syntax-based, and concern-based views. These views are aimed at tacking the models' complexity (which can hinder model comprehension) and the specific need that each stakeholder may have to understand the models, with different levels of abstractions and detail. We instantiate these views with *i** 1.0 models and introduce an implementation prototype with a modelling tool.
Project	NaPiRE – **Na**ming the **P**ain in **R**equirements **E**ngineering (NaPiRE) [151]
Summary	NaPiRE constitutes a globally distributed family of surveys on RE practices and problems. It started in 2012 by Daniel Méndez and Stefan Wagner, and since then it is conducted by an international group of researchers every 2 years. The main goal is to help the research community getting a better understanding of industrial trends in RE and problems faced therein. We are participating in the current edition, NaPiRE 2017, in particular in data collection (already finished) and data analysis (in progress). There are some conclusions from an initial analysis on the NaPIRE data. One key result is that companies use 4 (four) main documentation techniques: natural language, prototypes, user stories, and **use cases**. When further analysing the latter, we notice that use cases are highly used as the basis for the implementation, but also for tests, and in **customer acceptance** [152]. These results further emphasise the need for use cases specifications to have a good usability level, so that all stakeholders have a common understanding about these artefacts.

1.4 Document Outline

In Figure 1.1 we illustrate the organisation of this dissertation through a UML 2.0 activity diagram. This document is organised in 5 (five) parts. In the first part, we provide an

introduction and some technical background. It includes Chapters 1 and 2. The second part presents QualitEva, the proposed mixed-method process for the quality evaluation of requirements models, and the (bio)metrics used to perform that evaluation. It includes Chapters 3 and 4. As it can be observed in Figure 1.1, those two Chapters merge characteristics of background and research contributions. The reason for this combination is that the QualitEva process is an adaptation of Wohlin et al. guidelines [245] for experimentation in Software Engineering. Furthermore, and although several of the metrics are new and proposed as part of this dissertation, others are inherent to the biometric devices we used. The third part of this dissertation is related with the experimental work undertaken. It includes Chapters 5, 6, 7, and 8. The experimental reports presented in Chapters 6 and 7 can be visited in the the order the reader might prefer. However, Chapter 5 provides an insightful overview of the experiments and should be read as a base for the other two. The analysis presented in Chapter 8 is based on the data available in the previous Chapters, thus we recommend reading it afterwards. The fourth part of this dissertation contains the related work and it comprises Chapter 9. Finally, the fifth part of this dissertation contains the conclusions and points directions to future work, in Chapter 10.

More specifically, the document is organised as follows:

- **Chapter 2 – Background** presents a review of the main fields and topics in which this dissertation takes place. It includes information on requirements engineering, requirements representation and approaches, software quality with a focus on usability, and techniques for requirements quality evaluation.

- **Chapter 3 – QualitEva: A Mixed-Method Process for Quality Evaluation of Requirements Models** presents a step-by-step guide on how to perform (quasi-)experiments involving human subjects, and with the usage of (bio)metrics. It focuses on to the particular case of using (quasi-)experiments to evaluate the quality of requirements models through both the analysis of the models themselves, and the exploitation of human factors on how different people interact with them.

- **Chapter 4 – (Bio)Metrics for the Evaluation of Requirements Models** presents a set of (bio)metrics related with the evaluation of the *(i)* accuracy achieved by stakeholders when performing tasks on requirements models, as well as their *(ii)* speed *(iii)* visual ease; *(iv)* mental ease; *(v)* emotional ease; and *(vi)* perceived effort. We further propose metrics for the evaluation of *(vii)* *i** models.

- **Chapter 5 – A Family of 16 Quasi-Experiments for the Evaluation of Requirements Models** presents the experimental protocol followed in all the 16 quasi-experiments for the evaluation of the learnability and appropriateness recognisability of *i** (*i** 1.0 and iStar 2.0), and use cases (ARNE and ALCO templates).

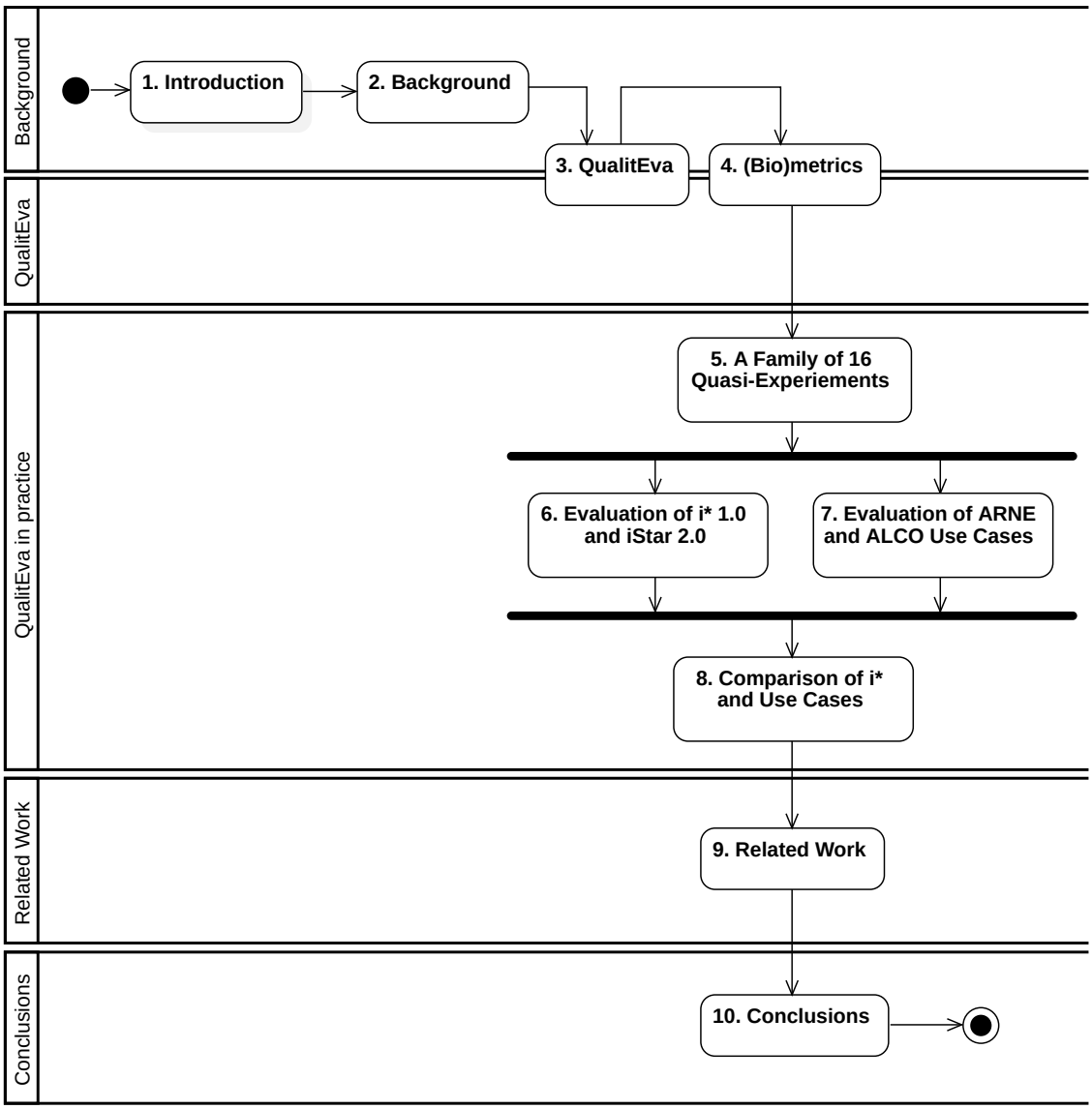


Figure 1.1: Dissertation outline.

- **Chapter 6 – Evaluation of *i** 1.0 and iStar 2.0** presents the experimental evaluation on the impact of different *i** versions (*i** 1.0 and iStar 2.0), as well as different individual characteristics, when participants are creating, modifying, understanding and reviewing *i** SR models.

- **Chapter 7 – Evaluation of ARNE and ALCO Use Cases** presents the experimental evaluation on the impact of different use case templates (ARNE and ALCO), as well as different individual characteristics, when participants are creating, modifying, understanding and reviewing use case specifications.

- **Chapter 8 – Comparison of *i** and Use Cases** presents a comparison of *i** 1.0, iStar 2.0, ARNE use case template, and ARCO use case template. This comparison is based on the data from the quasi-experiments reported in Chapters 6 and 7.

- **Chapter 9 – Related Work** presents a review of the state of the art in the fields and topics in which this dissertation takes place. Its main goal is to give the research context, describe the research topics, and further motivate the issues that were investigated in this dissertation.
- **Chapter 10 – Conclusions** presents the conclusions of this dissertation, with the answer to the research questions, a discussion on the contributions and limitations, as well as lessons learnt and ideas for further research.

In this dissertation, we describe various quasi-experiments with a great detail, to support their replication. Furthermore, we provide a complete replication package, available online, that does not require the reading of this document. To present further materials to support verification of the results and assist replications, we use a webpage [213]. We decided not to use the appendix, because we often have a large amount of tables and spreadsheets with information that cannot reasonably be presented in a text document. Furthermore, we avoid greatly increasing the number of pages.

Chapter 2

Background

This dissertation aims at performing a quantitative assessment of *i** 1.0, iStar 2.0, and ARNE and ALCO use cases' usability, in terms of *learnability* and *appropriateness recognisability*, by using a combination of (bio)metrics. Measurements and metrics are the basis of Empirical Software Engineering, an area in which experiments on software systems or artefacts are used to form and validate hypotheses about Software Engineering methods and techniques [60]. Software Engineering, however, is a vast engineering discipline, covering all aspects of software production, going from the early stages of system specification, until the maintainability of the system after it is being used in a real-world environment. In this dissertation, we focus on a subset of Software Engineering, namely: Requirements Engineering, requirements representation and approaches, software quality with a focus on usability, and techniques for requirements quality evaluation. In this Chapter, we review these fields and topics, with the objective of introducing the most important and relevant technical background.

2.1 Requirements Engineering

Requirements Engineering (RE) is the process of developing a software specification, by understanding and defining what services are required from a given software system, and by identifying all the constraints on that system's operation and development [39]. There are 5 (five) main activities in RE [126]:

- **Feasibility study** is the process of estimating whether the identified stakeholders' needs may be satisfied. The study considers if the proposed system will contribute to the organization goals, if it can be developed using current technology and within monetary constraints, and if it can be integrated with other systems. The result of the study informs the decision of whether or not to perform a more detailed analysis.

- **Requirements elicitation and analysis** is the process of discovering and reviewing the system requirements and constraints, by understanding stakeholders' needs. Technical personnel work directly with different stakeholders to identify the system domain, the services that should be available, and the operational constraints. This may involve the development of one or more system models and prototypes, in order to help understanding the system to be specified.

- **Requirements specification** is the process of documenting, in a clear and precise manner, the information gathered during the previous phase. Three types of requirements may be included in this document: user requirements (abstract statements of the system requirements for the end-user); system requirements (a more detailed description of the functionalities to be provided and, therefore, implemented); and domain requirements (which refer to the main domain concepts).

- **Requirements validation** is the process of ensuring that the requirements are complete, consistent and clear. During this process, errors, omissions and inconsistencies in the requirements document are discovered and rectified. This aims to ensure the quality of what was previously specified.

- **Requirements management** is the process of managing changes to requirements, ensuring that those changes are properly analysed and tracked throughout the system, and that their impact is well understood.

Other activities have been proposed (see Sommerville [211] or Lamsweerde [130]), depending on the type of system being developed, the system domain, the stakeholders, or the specific practices of the organisation itself. However, the activities are similar and, in all the cases, they are not performed in a strict sequence. In particular, requirements analysis continues during specification, and new requirements can appear throughout the process, so it is necessary to manage them. When there is a change in the requirements (new ones are added or old ones are removed), requirements engineers need to ensure that those changes are properly analysed and tracked throughout the software system, and that their impact is well understood. In this dissertation, we focus on requirements specification, validation and management.

2.2 Requirements Representation and Approaches

Natural language has been used to write software requirements since the beginning of software engineering [211]. Although it is expressive, intuitive, and universal (assuming stakeholders speak the same language), it is also potentially vague and prone to omissions and ambiguity. Its interpretation may also depend on the background of the readers. As a result, there have been proposals for alternative ways to represent requirements, namely through diagrammatic (visual) requirements models [170]. These models are used to

provide abstractions at some level of precision and detail. They are then analysed in order to obtain a better understanding of the software system [76] being developed, prior to its implementation.

Visual and textual representations of requirements can complement each other, and are often used together for requirements elicitation and analysis [101], where communication with different stakeholders plays a major role. For this communication to be effective, stakeholders need to have a common understanding of the requirements models [32].

In an effort for standardisation, and aiming at improving communication among the various stakeholders, different techniques or approaches for requirements specification and analysis have been proposed, namely:

- **Goal-oriented** [129] uses the notion of goal to elicit, elaborate, structure, specify, analyse, negotiate, document and modify software requirements. Goals are statements expressing properties that the software system must guarantee. They can be formulated at different levels of abstraction, making the specification of requirements more appropriate to the various stakeholders.

- **Agent-oriented** [249] uses the agent as the main abstraction, allowing modelling complex systems through a social metaphor. It allows us to capture characteristics as non-functional and organizational requirements in an explicit manner. It is often combined with goal-oriented requirements engineering.

- **Scenario-based** [211] uses examples of existing behaviours, the scenarios, to complement the information about the software system. Since it uses real examples, it makes it easier for different stakeholders to understand the requirements.

- **Object-oriented** [20, 118] uses objects to define the requirements of the software system being developed. The objects contain information about their functionality, their behaviour and their interactions with other objects.

- **Viewpoint-oriented** [126, 212] uses viewpoints to identify and organize the requirements according to different perspectives. A viewpoint represents a set of informations about the system, from the perspective of a particular stakeholder. Gathering different points of view enables the understanding of possible conflicts among them.

- **Aspect-oriented** [177] uses aspects to identify and specify cross-cutting concerns, such as availability and security. It is possible to identify the influences of these concerns on other concerns of the software system, reinforcing the modularity of this same system through mechanisms of abstraction, representation and composition.

In this dissertation, we have chosen 3 (three) of those approaches: **goal-oriented**, **agent-oriented**, and **scenario-based**. For the languages themselves, we selected *i** 1.0 and

iStar 2.0, combining goal and agent-oriented approaches; and use cases, for the scenario-based approach. The *i** framework is widely used in academia [97], with working groups in over 20 countries [243] and several editions of international workshops [241]. Use cases, in turn, are commonly used by practitioners, especially in their free textual form and text with constraints [151].

2.2.1 *i** 1.0 and iStar 2.0

The *i** framework, which we refer to as ***i** **1.0**, was developed for modelling and reasoning about organisational environments and their information systems [247, 248], covering both agent and goal-oriented modelling. It focuses on the concept of *intentional actor* and it is suitable for an early stage of the software system development, to better understand the problem domain. This framework has 2 (two) main modelling views:

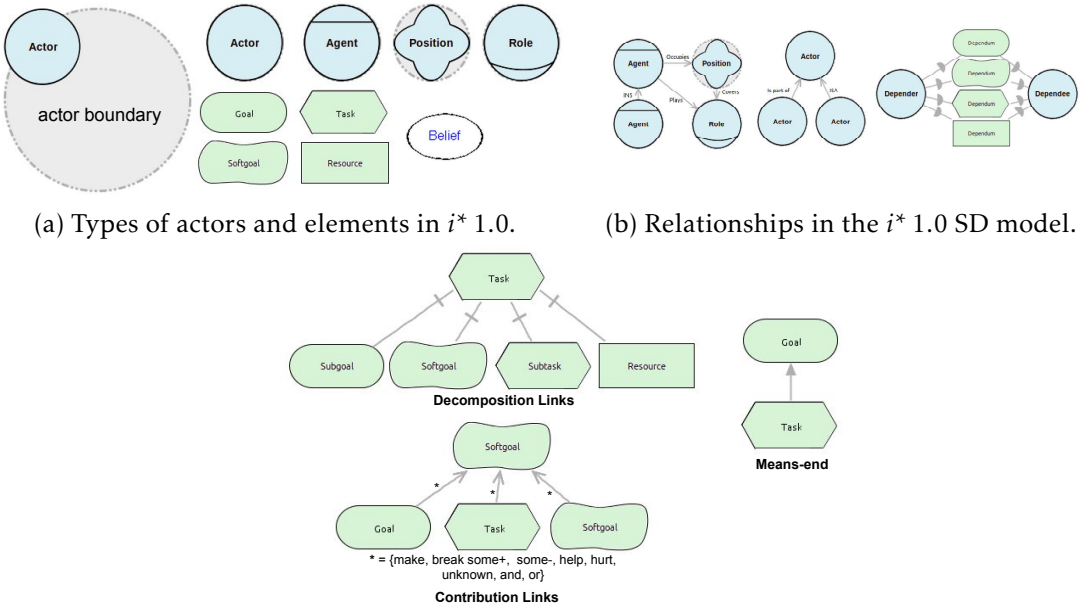
- **Strategic Dependency** (SD) model describes dependency relationships, through *dependency links*, among the actors in an organisational context. In this model, an actor (called *depender*) depends on another actor (called *dependee*) to provide an intentional element (called *dependum*). This intentional element can be a *task* to be performed, a *resources* to be obtained, or a *belief* to be expressed. The dependency links can have types, such as *open*, *committed* or *critical*.

- **Strategic Rationale** (SR) model provides more detail than the SD model, since it focuses on intentional elements and relationships internal to actors (represented inside the *actor boundary*). Intentional elements (goals, softgoals, tasks, resources and beliefs) are related by *means-end* or *decomposition links*. *Means-end* links can be perceived as decomposition links that are used to link *goals* (ends) to *tasks* (means) in order to specify alternative ways to achieve goals. *Decomposition links* are used to decompose tasks. A task can be decomposed into: a *subgoal*, a *subtask*, a *resource*, and/or a *softgoal*. Apart from these two links, there are the *contribution links*, which can be *positive* or *negative*, and are used to link intentional elements to *softgoals*.

In Figures 2.1a, 2.1b and 2.1c we illustrate the concrete syntax of *i** 1.0, as well as the relationships between the different elements for the SD and SR models, respectively.

Over the years, *i** 1.0 has been applied in many areas, such as healthcare [61], security [136] or e-commerce [37], and was subject to extensions and variations, like Tropos [73] and GRL [135]. Despite its larger academic adoption, the diversity of extensions and variations can make it difficult for novices to learn and use it in a consistent way [78]. For this reason, **iStar 2.0** was created, evolving *i** 1.0 into a consistent and clear set of core concepts [49]. In Figure 2.2 we represent the differences introduced by iStar 2.0, including new relationships to replace some of the previous ones.

In order to better summarise the evolution of the *i** framework, in Table 2.1 we present a comparison between *i** 1.0 and iStar 2.0, with the corresponding changes. As it can be



(a) Types of actors and elements in *i** 1.0.

(b) Relationships in the *i** 1.0 SD model.

(c) Relationships in the *i** 1.0 SR model.

Figure 2.1: Notation and allowed model relationships of *i** 1.0.

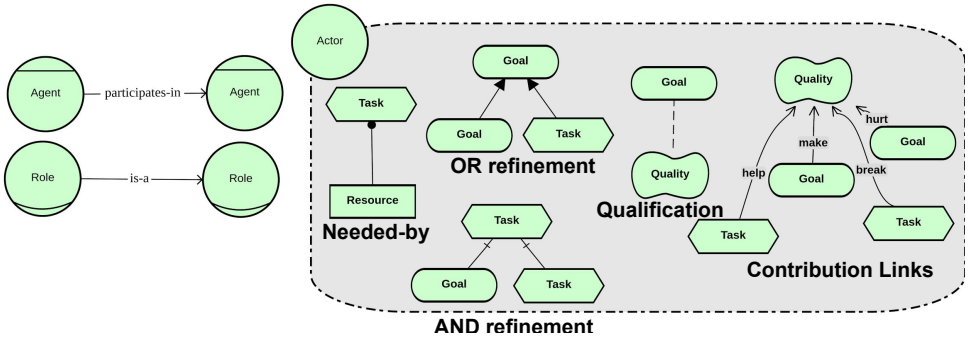


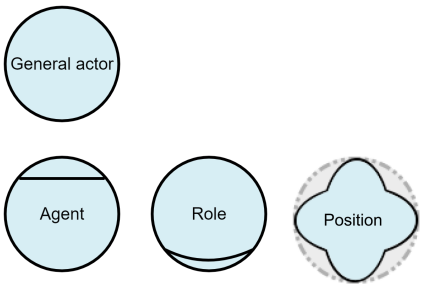
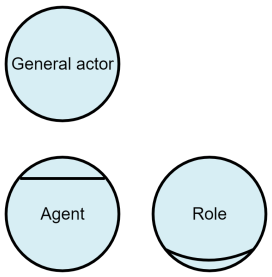
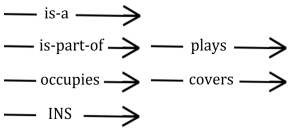
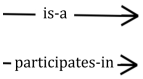
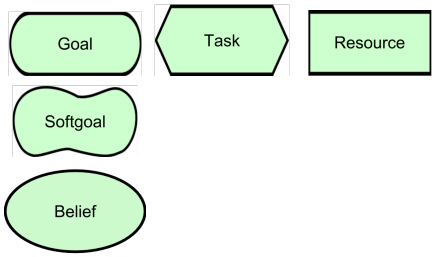
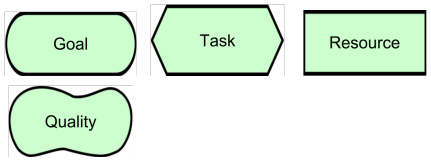
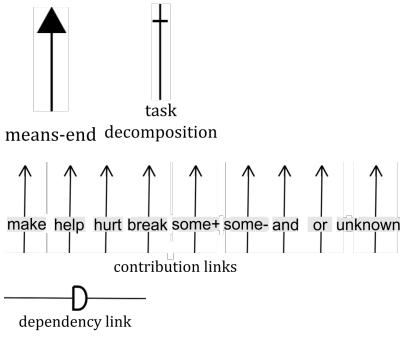
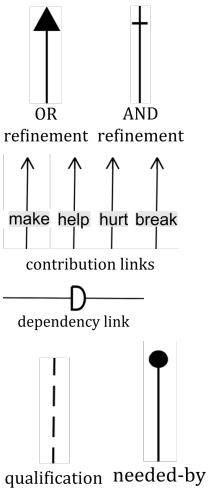
Figure 2.2: Notation of the iStar 2.0 and allowed relationships between model elements.

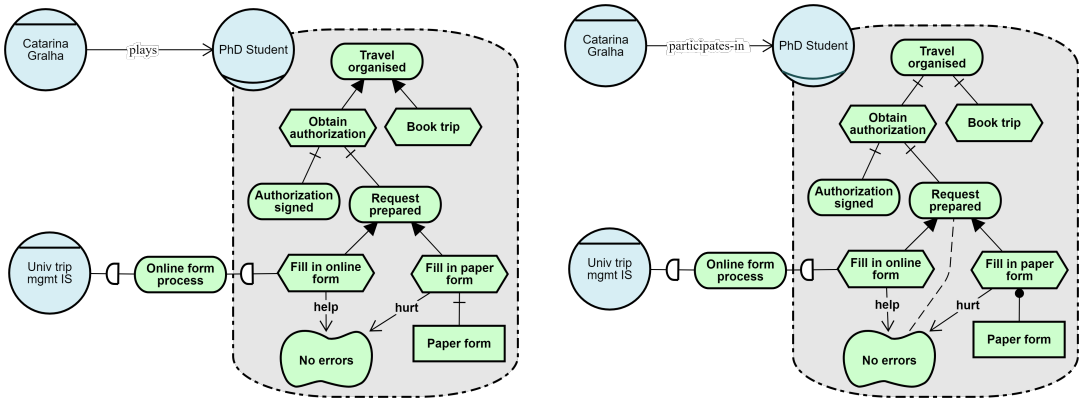
observed, some differences where introduced by iStar 2.0, including the removal of some model elements, and new relationships to replace some of the previous ones. The type of actor *position* was removed and the actor links were greatly simplified. The element *belief* was also removed from the language, and *softgoals* were named *qualities*. The *means-ends link* and the *task decomposition link* were redefined, and named *OR* and *AND refinement*, respectively. The *contribution links* were simplified and 2 (two) new links were added: *needed-by* and *qualification*.

To illustrate the application of some of these concepts, in Figure 2.3 we show a PhD student that wants to organise trips to conferences and relies on the university's trip management information system, in both *i** 1.0 (Figure 2.3a), and iStar 2.0 (Figure 2.3b).

The *i** standardization process is not yet concluded, and there is a need for studies

Table 2.1: Comparison between *i** 1.0 and iStar 2.0, adapted from [132].

	***i** 1.0**	**iStar 2.0**
Actors		
Actor links		
Intentional elements		
Intentional element links		



(a) i^* 1.0 model example. (b) iStar 2.0 model example.

Figure 2.3: A PhD student wants to organize trips to conferences, adapted from [49].

about iStar 2.0 ease of use, adequacy for teaching, expressiveness, and graphical notation [49], which is in line with the goals of this dissertation.

2.2.2 Use Cases

A **use case** defines a sequence of interactions between one or more actors and a software system [110, 111]. Actors are the only external entities that interact with the system, which means that they are outside the software system and are not part of it. There are 2 (two) types of actors: primary and secondary. A **primary actor** initiates a use case to which the system has to respond. **Secondary actors** participate in a use case started by the primary actor [76]. Whereas a simple use case might involve only one interaction, a more typical use case will consist of several interactions.

When a use case gets too complex, parts of it can be split into other, separate, use cases. A common scenario is when a use case fragment is duplicated and appears in multiple use cases. This fragment can become a use case by itself, and dependency relationships between the original and this new use case must be specified, by using *includes* or *extends* relationships. The goal is to maximize extensibility and reuse of use cases.

There are a variety of templates to textually write use cases, which help writing more structured specifications [2]. Constraining the way the use cases are written is important for improving their readability, understandability, consistency and completeness, since it may help reducing the ambiguity and omissions introduced by the use of natural language.

In this dissertation, we are using 2 (two) different use cases templates: the one proposed by **Ar**low and **Ne**ustadt [5], which we named **ARNE template**; and the one proposed by **Al**istair **Co**ckburn [42], which we named **ALCO template**. The former is the template used at the Software Development Methods course at our University, which is simpler but highly used both in academia and in industry. The latter is one of the most complete use cases templates, and it is also widely known and accepted [226]. The goal

is to contrast a simpler with a more detailed use case specification. In Tables 2.2 and 2.3 we present the basic concepts of ARNE and ALCO templates, respectively.

Table 2.2: Basic concepts of the ARNE use case template [5].

Name	*Name for the use case.*
Brief description	*Paragraph that summarises the goal of the use case.*
Actors **Primary** **Secondary**	*Entities participating in the use case.* *Is responsible for initiating the use case.* *Interacts with the use case after it is initiated.*
Pre-conditions	*Constraints on the state of the system before the use case can start.*
Main flow	*Lists the steps in a use case that capture the situation where everything goes as expected, and there are no errors, deviations, interrupts or branches. It always begins with the primary actor doing something.*
Post-conditions	*Constraints on the state of the system after the use case has executed.*
Alternative flows	*List of alternatives to the main flow. They can capture errors, branches, and interrupts to the main flow.*

Table 2.3: Basic concepts of the ALCO use case template [42].

Name	*Name for the use case.*
Context of use	*Longer statement of the goal, if needed, its normal occurrence condition.*
Scope	*Design scope, what system is being considered black-box under design.*
Level	*Can be one of: strategic, user goal, subfunction.*
Primary actor	*Role name for an entity responsible for initiating the use case.*
Stakeholders & interests	*List of stakeholders and key interest in the use case.*
Pre-conditions	*What we expect is already the state of the world.*
Success end condition	*The state of the world upon successful completion.*
Failed end protection	*The state of the world if the goal is abandoned.*
Trigger	*The action that starts the use case. It can be a time event.*
Main success scenario	*Lists the steps of the scenario from trigger to goal deliver.*
Variations	*Branching actions, things that will cause eventual bifurcation in the scenario.*

Use cases are not only represented by text, but also by diagrams, named **use case models**. While use cases specifications detail each use case interaction, use cases models show only the main features and their relations with actors. However, they complement each other, being an effective tool for a better understanding, communication and design of complex system behavioural requirements [76]. Nonetheless, use case models are **not** covered in the context of this dissertation.

2.3 Software Quality

To the best of our knowledge, there is no ISO standard for the quality of requirements models. The ISO/IEC/IEEE 29148:2018 [108] is related to the engineering of requirements for systems and software products, and defines the construct of the good requirements and provides attributes and characteristics of requirements. However, it does not provide quality measures for quantitatively evaluating system and software product quality. This is defined the by ISO/IEC 25023:2016 [107]. As such, in this dissertation we opted for using the ISO/IEC 25023:2016 instead of the ISO/IEC/IEEE 29148:2018.

Quality, in the context of software systems, is the degree to which a system, component or process meets the specified requirements, that is, the degree to which it satisfies the stated and implied needs of its stakeholders, and thus provides value [107]. As illustrated in Figure 2.4, **quality** is divided into 8 (eight) characteristic:

- **Functional suitability** is the degree to which a product or system provided functions that meet stated and implied needs when used under specified conditions.
- **Performance efficiency** is the degree of performance relative to the amount of resources used under stated conditions.
- **Compatibility** is the degree to which a product, system or component can exchange information with other products, systems or components, and/or perform its required functions, while sharing the same hardware or software environment.
- **Usability** is the degree to which a product or system can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use.
- **Reliability** is the degree to which a system, product or component performs specified functions under specified conditions for a specified period of time.
- **Security** is the degree to which a product or system protects information and data so that persons or other products or systems have the degree of data access appropriate to their types and levels of authorization.
- **Maintainability** is the degree of effectiveness and efficiency with which a product or system can be modified by the intended maintainers.
- **Portability** – the degree of effectiveness and efficiency with which a system, product or component can be transferred from one hardware, software or other operational or usage environment to another.

In this dissertation, we are studying the **usability** of requirements models. For that reason, and for the sake of brevity, usability is the only characteristic that has its 6 (six) sub-characteristics presented:

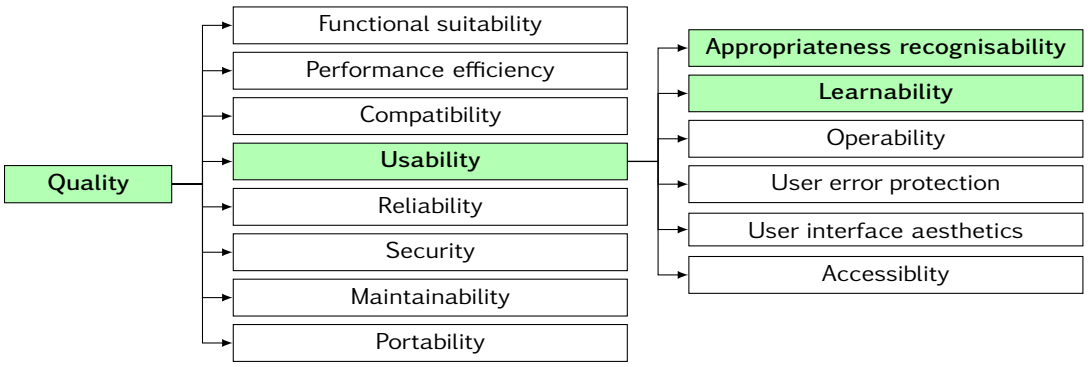


Figure 2.4: Quality characteristics according to the ISO/IEC 25023:2016 [107].

- **Appropriateness recognisability** *(previously known as understandability)* is the degree to which users can recognize whether a product or system is appropriate for their purposes and particular tasks.
- **Learnability** is the degree to which a product or system can be used by specified users to achieve specified goals of learning, to use the product or system with effectiveness, efficiency, freedom from risk and satisfaction in a context of use.
- **Operability** is the degree to which a product or system has attributes that make it easy to operate and control.
- **User error protection** is the degree to which the product or system protects users against making errors.
- **User interface aesthetics** is the degree to which the user interface enables pleasing and satisfying interaction for the user.
- **Accessibility** is the degree to which a product or system can be used by people with the widest range of characteristics and capabilities to achieve a specified goal in a specified context of use.

Since one of the objectives of this dissertation is to analyse whether and to which extent different stakeholders are able to create, modify, understand and review requirements models, we are interested in **appropriateness recognisability** and **learnability**. In the previous ISO/IEC 25023:2016 [107] definitions, learnability is the only sub-characteristic of usability that contemplates *freedom from risk*. However, it is not present in the definition of usability itself. As such, freedom from risk is **not** addressed in this dissertation.

2.4 Requirements Quality Evaluation

2.4.1 The Goal-Question-Metric Approach

The Goal-Question-Metric (GQM) approach [8] is a metrics' definition technique recommended by the IEEE Computer Society [104], used as a standard by the Empirical

Software Engineering community. It is a goal-oriented approach for defining and interpreting software measurements, having the following 3 (three) levels of abstraction:

- **Conceptual Level (Goal)** – a set of goals is defined, by listing the main objectives of the software project.
- **Operational Level (Question)** – a set of questions is used to characterise the way the achievement of a specific goal is going to be performed.
- **Quantitative Level (Metric)** – a set of metrics is associated with every question in order to answer it in a quantitative way.

A GQM model is a hierarchical structure, as we present in Figure 2.5, that starts with a goal or set of *goals* (**conceptual level**), specifying purpose of measurement, object and issue to be measured, and viewpoint from which the measure is taken. Each goal is refined into *questions* (**operational level**) that usually break down the issue into its major components, characterising how a goal can be achieved. Each question is then refined into *metrics* (**quantitative level**), which provide quantifiable information to answer the questions. Those data can be objective, if they depend only on the object that is being measured; or subjective, if they depend on the object and on the stakeholder's viewpoint. The same metric can be used to answer different questions under the same goal.

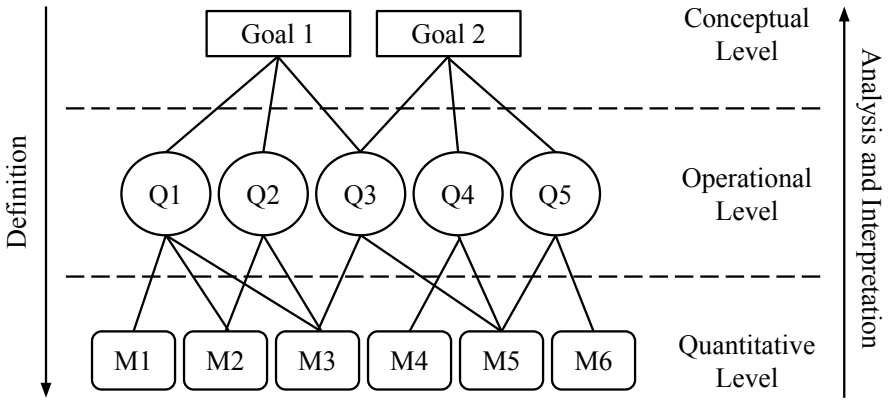


Figure 2.5: Hierarchical structure of GQM models

In this dissertation, we have used the **GQM approach** for defining metrics to evaluate the quality of requirements models in an objective, systematic and quantitative manner.

2.4.2 Requirements Metrics

Software quality metrics are a quantitative measure of the degree to which a software item or artefact possesses a given quality attribute [70]. Metrics can be classified into: (inside-out) product metrics and (outside-in) process metrics [117]. Requirements quality metrics, in particular, are a subset of process metrics, and can help in understanding

and improving the requirements management process and products, including the requirements models themselves. When incorporated in the requirements gathering and elicitation process, these metrics assist in analysing the quality of requirements models, as well as identifying the reasons for software problems [70].

Metrics can be specified both informally and formally [117]. An informal way of defining metrics is through natural language. As discussed in Section 2.2, natural language is simpler and easier to understand than a formal language. However, it is more prone to ambiguities and omissions, fostering inconsistent implementations. In turn, a formal specification of metrics avoids ambiguities that could result in inconsistent implementations of the metrics [203], and enables their automatic collection.

In this dissertation, we have used **Object Constraint Language (OCL)** [159, 239] to formally specify metrics. OCL is a declarative language used to describe expressions in UML (Unified Modelling Language [160]) models. These expressions typically specify invariant conditions that must be guaranteed in the modelled system, or queries about objects described in the model. When expressions in OCL are evaluated, they have no side effect, meaning that their evaluation can not change the state of the system where they are being executed. However, OCL expressions can also be used to specify operations or actions that, when executed, change the state of the system [159]. OCL is a precise textual language offering expressions that are free from the ambiguities of natural languages, and allows the expression of constraints in an object-oriented model that can not be specified through the diagram itself. It can be used for different purposes, such as specifying invariants in classes and types in the class model, to describe pre and post conditions in operations and methods, to specify constraints on operations, among others.

OCL has 4 (four) primitive data types: boolean, integer, real, and strings. In addition, it has logical operators (>, <, =, >= and <=), and the statements are built in 4 (four) parts:

- **Context** defines the limit situation in which the statement is valid.
- **Property** represents some features of the context (for example, if the context is a class, the property can be an attribute).
- **Operation** can be arithmetic or set-oriented; manipulates or qualifies a property.
- **Keywords** are used to specify conditional expressions (*if*, *then*, *else*, *and*, *or*, *not*, *implies*).

To better understand these concepts, in Figure 2.6 we illustrate a class diagram, with the classes: `Person`, `Company`, Job and `Marriage`.

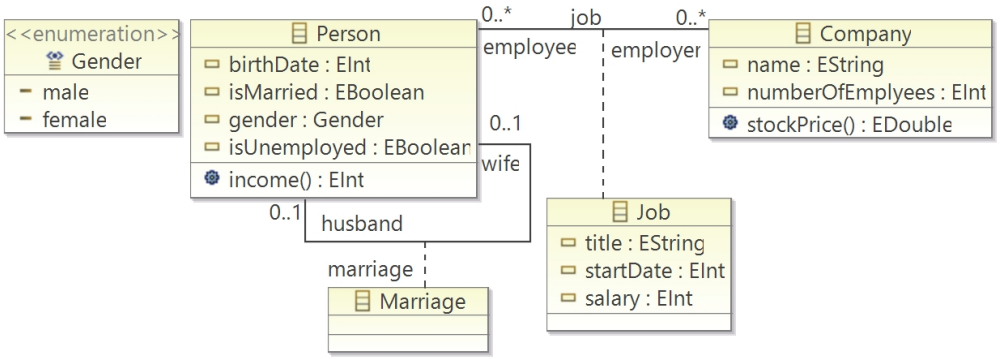


Figure 2.6: Example of a class diagram [239].

We intend to add the following restrictions to this diagram: (*i*) the number of employees in the company has to be greater than 50; (*ii*) the number of employees in a given job must be greater than 1 and their age must be greater than 18 years, we want to know who is the wife of a certain person; and (*iii*) if a person is unemployed, the corresponding salary is less than 100, otherwise it is equal to or greater than 100. In Listing 2.1 we present the OCL code for the previously specified constraints and operations, following the order.

Listing 2.1: Example of OCL code [239].

```
context Company
 inv: self.numberOfEmployees > 50

context Job
 inv: self.employer.numberOfEmployees >= 1
 inv: self.employee.age >= 18

context Person::getCurrentWife() : Person
 pre: self.isMarried = true and self.gender = male
 post: result = self.marriage.wife

context Person inv:
 let income : Integer = self.job.salary -> sum() in
  if isUnemployed then
   income < 100
  else
   income >= 100
  endif
```

OCL was chosen as the formal language used in this dissertation since it offers formality without sacrificing understandability, given it was developed considering its usability for UML professionals [6]. Furthermore, since the definitions are executable, it is possible to avoid accuracy problems in the metrics implementation, which often occur with other approaches [80]. Moreover, it is a flexible approach, and to add a new metric we only need to define a new OCL rule, specifying how the metric should be computed. Finally,

it leads to a direct integration with the meta-model, facilitating the process of metrics definition.

2.4.3 Eye-tracking Technology

Eye-tracking is a technology that measures the activity of the eyes. In human vision, eye-movements are essential to collect evidence regarding participants' cognitive processes [175, 193]. Where do we look? What do we ignore? How do we analyse visual information? In order to answer those questions, eye-tracking devices, called **eye-trackers**, monitor a participant's visual attention by collecting eye-movement data when (s)he looks at a stimulus, while working on a specific task. A stimulus is an object, such as text, source code, or diagram, that is necessary to perform that task [55, 178].

Beyond the analysis of visual attention and cognitive processes, eye-data can also be examined to measure the workload of a task. Furthermore, the data can be studied with respect to certain areas of the stimuli, which are called **areas of interest** (AOI). An AOI can either be relevant or irrelevant for a given task that is being performed by a participant. For example, when considering a class diagram as stimulus, a relevant AOI could be a specific class that is used by the participant to perform a certain task, while an irrelevant AOI would be any other classes in the diagram [193].

There are many methods for exploring eye-data. The most common one is to analyse the visual path of the participant across a computer screen, where each eye-data observation is translated into pixel coordinates. From there, the presence or absence of eye-data points in different AOIs can be examined. This type of analysis is used to determine which features are seen, when a particular feature captures attention, how quickly the eye moves, what content is overlooked, among other gaze-related questions. Typically, eye-data is classified based on 3 (three) indicators [193]:

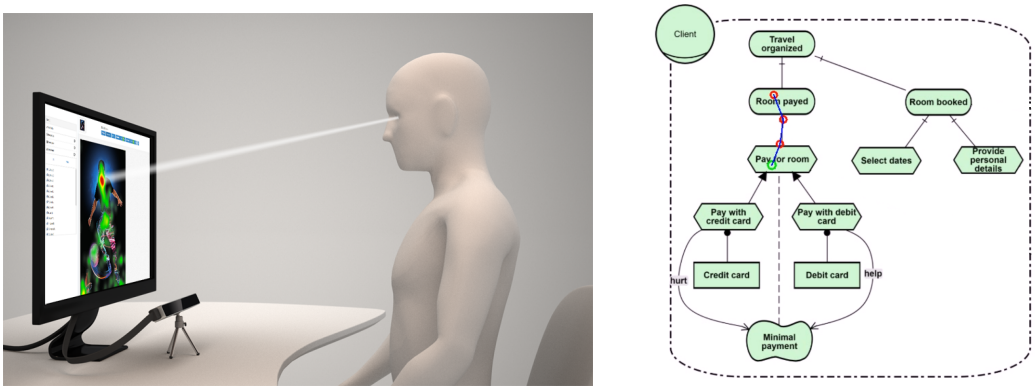
- **Fixation** is a stabilization of the eye on a part of the stimulus for a period of time between 200 and 300 ms. Most of information acquisition and cognitive processing occur during fixations, but only a small set of them is needed for participants to acquire and process a complex visual input [172].
- **Saccade** is a sudden and quick eye-movement from one fixation to another, lasting between 40 to 50 ms. Information encoding and cognitive processing that occur during a saccade is very limited [55, 172].
- **Scan-path** is a series of fixations in chronological order, representing the tasks performed by participants [172]. An AOI is visited if it had at least one fixation in it.

Eye-trackers can be intrusive or non-intrusive. The first generation of **intrusive eye-trackers** typically contains miniature cameras that are mounted on a padded headband, that participants wear during the studies. Two of these cameras capture eye-movements

using infrared lights reflecting on the participants' pupils, while the third one is optional and used for head tracking. The second generation devices are similar to normal glasses. The first generation of **non-intrusive eye-trackers** normally uses beams of light that are reflected on the participants' eyes. Such eye-trackers have have low resolutions and precisions [193]. Those of the second generation are called video-based eye-trackers, and generally consist of one computer, one or two cameras, and one infrared pad, which can be integrated with the cameras. The computer tracks the participants' eye-movements by detecting the positions of the participant's head using eye-brows, nose, and lips.

In all the eye-trackers, the eye-data is written to a file. Normally, the file is created automatically and it is compatible with an eye-tracking analysis software, provided with the eye-tracker. However, with some low cost eye-trackers, an Application Programming Interface (API) is available, but the software needed to collect and analyse the eye-data must be developed by the researcher.

In this dissertation, we have used **The Eye Tribe** [222] (see Figure 2.7a). We opted by a non-intrusive device to mitigate the threat of validity that a more intrusive eye-tracker places in a cognitive task. We developed a custom software to collect the data from this eye-tracker, in collaboration with two exchange students, Arkadiusz Karbowy and Łukasz Golebiowski, and one masters' student, Mafalda Santos. We collect: a time stamp and the x and y pixel coordinates of the gaze; if a fixation was detected; the duration of that fixation; and pupils dilatation. Those data are written to a CSV file, and a video is recorded with the real-time eye trajectory and the detected fixations (see Figure 2.7b).



(a) User in front of The Eye Tribe [223]. (b) Video frame with the eye-data.

Figure 2.7: Eye-data is captured by the Eye Tribe Tracker and recorded in video: a line with the eye trajectory, fixations in the first 3 circles, and current fixation of the eye in last circle.

After the collection of eye-data, it is necessary to analyse it, in order to obtain quantitative measures. There are several metrics and visualisation techniques that can be used. For organisation purposes, we present them in Chapter 4.

2.4.4 Electroencephalography (EEG) Scanners

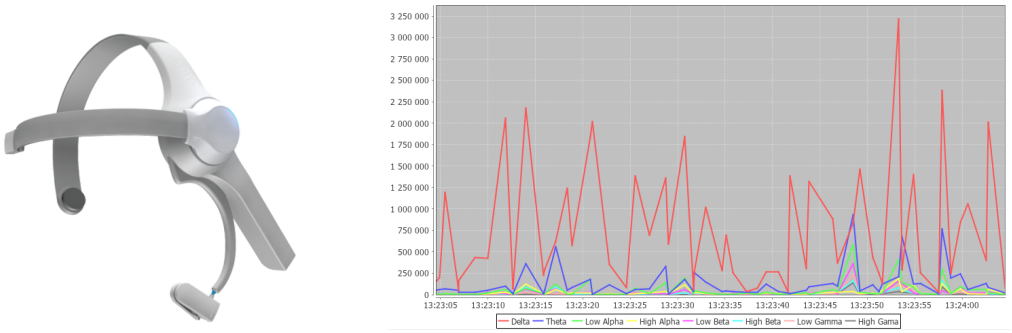
Electroencephalography (EEG) refers to the measurement of the brain's electrical activity that arises from neuronal firing. The varying activity of neurons in the brain causes fluctuations in the voltage potential along the scalp that can be measured with an **EEG scanner** [3]. When analysing EEG data, the focus is generally on the spectral content of the EEG, that is, the type of neural oscillations (also known as brain waves) that can be observed. Brain waves can be divided into frequency bands, called alpha (α, 8-12 Hz), beta (β, 12-30 Hz), gamma (γ, 30-100+ Hz), delta (δ, 0-4 Hz), and theta (θ, 4-7 Hz) [92].

Although EEG scanners started by being used to diagnose epilepsy, sleep disorders, coma, encephalopathies, and brain death [221], some work has linked these specific frequency bands with mental workload, task engagement and emotions [127, 133, 150]. Each of the frequency bands has a specific frequency range and amplitude, exhibiting more or less activity under certain circumstances. For instance, alpha waves can typically be observed when an individual is in a relaxed state, but they either disappear or their amplitude decreases significantly as soon as the physical or mental activity increases [3].

EEG scanners can be invasive or non-invasive. **Invasive** scanners are generally used in medicine and are made with electrodes that have been surgically implanted on the surface or within the depth of the brain. **Non-invasive** can be wet or dry. Wet EEG scanners consist on a cap that is placed on the head and where the electrodes present on the cap require a conductive gel. Dry EEG scanners are smaller, more portable, and easier to use. They are also called **wearable EEG**, and are based upon creating low power wireless collection electronics and dry electrodes, not requiring a conductive gel [36].

In this dissertation, we have used **NeuroSky MindWave Headset** [156], a non-invasive EEG scanner (see Figure 2.8a). It is a one-channel, noise-filtering, dry sensor that records the EEG signal at 512 Hz from a single location on the user's forehead, reading signals mainly from the pre-frontal cortex. The scanner is also highly sensitive to the motor signals of the face, such as brow furrowing, blinking, and eyebrow motion. Each of these motor activities produces a high amplitude, low frequency signal which is easy to distinguish from neuronal activity. We developed a custom software to collect the data from this EEG scanner. We collect a time stamp and brain waves (frequency bands: alpha, beta, theta, gamma, delta and theta). Moreover, we compute attention, mental workload and familiarity scores. Those data are written to a CSV file, and a chart is generated in real-time with the frequency bands collected (see Figure 2.8b).

After the collection of brain's electrical activity, it is necessary to analyse it, in order to obtain quantitative measures. NeuroSky MindWave has pre-built metrics and algorithms, categorized into attention, meditation, blink detection, mental effort, familiarity, appreciation, emotional spectrum, cognitive preparedness, creativity and alertness. For organisation purposes, we present them in Chapter 4.



(a) The NeuroSky MindWave.

(b) Chart with the EEG frequency bands.

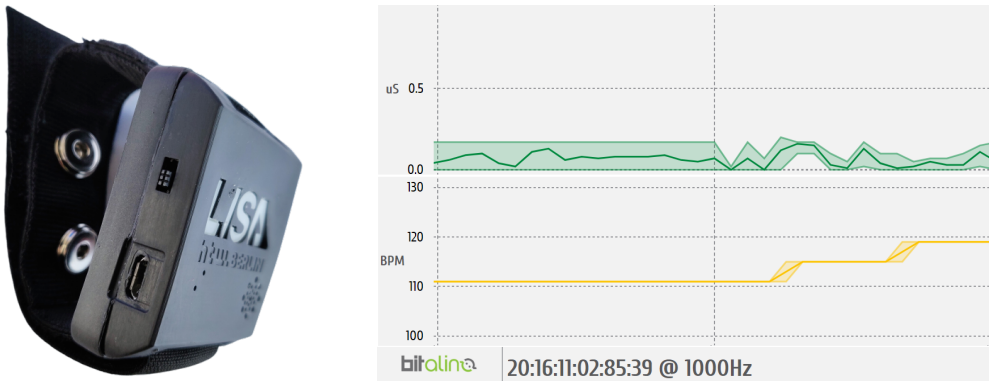
Figure 2.8: EEG-data is captured by the NeuroSky MindWave Headset and recorded in a file, while a chart of the frequency bands is generated: *delta* waves are associated with attention, *theta* with resting, *alpha* with mediation, *beta* with active thinking or focus, and *gamma* with memory and recognition.

2.4.5 Electrodermal Activity (EDA) Scanners

Electrodermal activity (EDA) is a biological property of the human body that causes continuous variation in the electrical characteristics of the skin, being an output of the sweat glands on a microscopic level. The EDA signal is characterised into 2 (two) types: the tonic skin conductance level, which has low frequency and changes slowly; and phasic skin conductance response, which has a high frequency and changes fast [190]. Sweating is controlled by the sympathetic nervous system [138] and if the sympathetic branch of the autonomic nervous system is highly aroused, sweat gland activity increases, which in turn increases skin conductance. **EDA scanners** measure this electrical skin conductance, which serves as an indicator for emotional stimuli [58, 89]. When an individual experiences emotional activation (such as excitement or stress), or an increased cognitive workload, or physical exertion, the brain sends signals to the skin to increase the level of sweating. One may not feel any sweat on the surface of the skin, but the electrical conductance increases in a measurably significant way as the pores begin to fill below the surface [34].

In this dissertation, we have used the **BioSignalsPlux Wristband [17] with BITalino [18]**, a custom-made EDA scanner (see Figure 2.9a). The scanner also has a photoplethysmogram (PPG) sensor, which detects blood volume changes in the microvascular bed of tissue, and is used to determine heart rate variability. An increase in the heart rate, when in a stationary state, can be related with nervousness or anxiety [54, 137] and mental stress [208]. A software is provided with the EDA scanner, **OpenSignals** [161], and it collects a time stamp and the bio-signals. Those data are written to a file, and a chart is generated in real-time with the frequency waves (see Figure 2.9b).

After the collection of skin's electrical activity and heart rate, it is necessary to analyse it, in order to obtain quantitative measures. OpenSignals has a built-in suite of signal processing and reporting add-ons, which enable data analysis and feature extraction. The



(a) The BioSignalsPlux. (b) Chart with the frequency waves from OpenSignals.

Figure 2.9: EDA-data is captured by the BioSignalsPlux Wristband and recorded in a file, while a chart of the frequency waves is generated: electrical skin conductance (in μS) is associated with cognitive workload and, heart rate variability (in BPM, Beats Per Minute) is associated with nervousness.

algorithms are proprietary, and the final user only has access to the final report. For that reason, it is not possible to provide further details about the algorithms and the statistical methods applied. However, the metrics used in the algorithms are known. For organisation purposes, we present them in Chapter 4.

2.4.6 Subjective Workload and Cognitive Load Assessment

Cognitive load can be defined as a multi-dimensional construct representing the load that a task imposes on a participant [162, 163]. This also refers to the level of perceived effort for learning, thinking and reasoning as an indicator of pressure on working memory during task execution [246]. This measure of mental workload represents the interaction between task processing demands and human capabilities or resources [91, 240]. In the 1980's, several techniques for subjective workload and cognitive load assessment were proposed [95, 246]. However, only a few are frequently used, such as the NASA-Task Load Index (NASA-TLX), and the Subjective Workload Assessment Technique (SWAT) [90, 95].

The NASA-TLX [93, 94] uses 6 (six) dimensions to assess workload and cognitive load: mental, physical, and temporal demand, performance, effort, and frustration. Twenty-step bipolar scales, shown in Figure 2.10, are used to obtain ratings for these dimensions, and a score from 0 to 100 is obtained on each scale. Then, a weighting process with a paired comparison is used: the participant chooses which dimension is more relevant to workload for a particular task across all pairs of dimensions. The number of times each dimension is chosen is the weighted score. This is multiplied by the scale score for each dimension and then divided by 15 to get a workload score from 0 to 100.

The SWAT [179] uses 3 (three) dimensions to assess workload and cognitive load: time load, mental effort load, and psychological stress. For each dimension, the participant

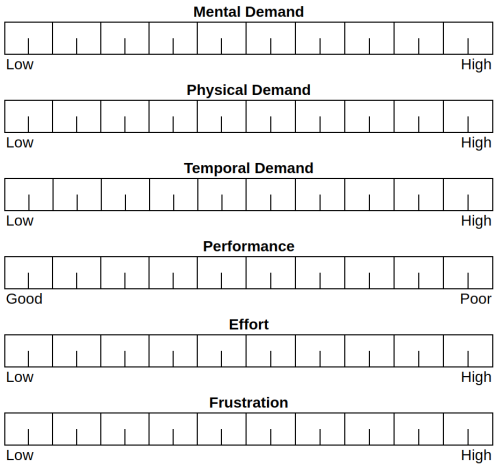


Figure 2.10: The 6 dimensions of NASA-TLX [94].

selects low, medium or high. Then, the measurement scores are scaled to produce a single rating scale with interval properties.

When the SWAT is compared to the NASA-TLX, there is a very high convergent validity between them [182]. However, the latter is generally considered to be the better scale for measuring mental workload, since it is slightly more sensitive in terms of the dimensions associated with mental load [95]. Moreover, NASA-TLX shows a higher correlation with performance [182]. For these reasons, we decided to use **NASA-TLX** in this dissertation.

2.4.7 Gender-specialised Cognitive Assessment

Research into gender differences has determined that individual characteristics in how people solve problems often cluster by gender [10, 191]. In software systems, it is common to have features that are inadvertently designed to be more supportive of problem-solving processes typically followed by males than by females [86, 219].

Awareness of these gender biases within software systems has increased [69, 202]. Addressing this problem and designing software systems to be more gender-inclusive can benefit all problem solvers, regardless of their gender [114, 220]. In order to help software practitioners evaluate their software system from a gender-inclusiveness perspective, GenderMag (**Gender** Inclusiveness **Mag**nifier) [27] was created.

GenderMag can be described as an analytical method for evaluating usability with a focus on gender-inclusiveness. It has 5 (five) problem-solving facets that have been extensively investigated in the literature: (*i*) motivation for using the software, (*ii*) information processing style, (*iii*) computer self-efficacy, (*iv*) attitude towards risk, and (*v*) ways of learning new technology. GenderMag proposes 4 (four) personas to bring those facets to life: Tim, Abby, Pat(ricia) and Pat(rick) (we will refer to them as Pats).

- **Tim** has facet values most frequently seen in males, that are most different from those seen in females. As such, Tim represents a large fraction of males (as well as

a few females).

- **Abby** has facet values frequently seen in females, that are most different from those seen in males. Thus, Abby represents a large fraction of females (as well as a few males). Abby is meant to represent the *opposite* of Tim in terms of the 5 facets.

- **Pats** are identical except for their gender. They are a combination of facet values often occurring for females, with those somewhat less often occurring for females, and with those often occurring with both groups. They aim to show that differences relevant to inclusiveness lie in the facets, and not in a person's gender identity.

A persona is a vivid description of a given subset of a population, including their goals, motivations and attitudes [1]. In GenderMag, each persona has a value for every facet, and a specific background consistent with those facet values. In Table 2.4, we present the summary of the facet values for each persona. A complete characterisation of the personas is available at the GenderMag Project website [26].

Table 2.4: Summary of the facet values for each persona.

	Abby	**Pats**	**Tim**
Motivation	Technology is used to accomplish tasks	Technology is used to accomplish tasks	Technology is a source of fun
Information processing	Comprehensive	Comprehensive	Selective
Self-efficacy	Low compared to peer group	Medium	High compared to peer group
Risk	Risk-averse	Risk-averse	Risk-tolerant
Learning style	Process-oriented	Tinkering (reflectively)	Tinkering (sometimes excessively)

In this dissertation, rather than using the personas to define how requirements model should support the different facets, we have used a **GenderMag** questionnaire [236] to characterise stakeholders and determine their five facets levels. We are then able to explore how differences in the stakeholders facets influence the way they create, modify, understand and review requirements models.

2.4.8 Discussion

There are several ways to evaluate the quality of a software artefact, being the collection of metrics one of the most used methods [70]. Given that, one may ask: why not just collecting and analysing metrics? Whereas this analysis is useful for understanding requirements models, it may not be enough since it does not provide us insights about the interaction between stakeholders and the model. To do so, we can measure the success on tasks such as creating, modifying, understanding, and reviewing models, by collecting (*i*) direct task performance metrics such as precision, recall, and duration of those tasks;

and (*ii*) indirect measures such as the effort while performing them, assessed with biometric devices, and the participants' perceptions on their effort, measured with NASA-TLX. By characterising our participants with GenderMag, we are able to understand how each of the 5 (five) facets may impact the way people tackle the proposed tasks.

Although eye-trackers can give insights into where a subject is directing his eyes at a given time and how eye-movements are modulated by visual attention, tracking gaze positions alone does not inform about cognitive processes and the emotional states that guided the eye-movements. Eye-trackers can be complemented by other biometric sensors, such as EEG and EDA scanners. As seen in Subsections 2.4.4 and 2.4.5, EEG measures mental effort, in terms of concentration; and EDA measures the stress level. By using these equipments, we capture a broader view of the human behaviour in a given moment, gaining meaningful insights into the dynamics of attention, motivation, and emotion. We have chosen the particular sensors described in this Chapter (The Eye Tribe, NeuroSky MindWave Headset, and BioSignalsPlux Wristband with BITalino) for the following main reasons: (*i*) existing literature and research has linked the measurements recorded with these sensors to cognitive states and process, as well as emotions (see Chapter 4 for more details); (*ii*) these sensors are less invasive than other similar devices; and (*iii*) the sensors are affordable for an individual researcher or developer.

The combination of all these techniques gives us a multi-perspective about the models, the problems they may have, and the way stakeholders interact with them. Furthermore, by only using one method, we would be introducing a construct validity threat, the **mono-method bias**. Using a single type of measure introduces a risk: if there is a measurement bias, then the experiment will be misleading. By using different types of measures, they can be cross-checked against each other, producing stronger and more solid results [245].

2.5 Summary

In this Chapter, we started by presenting RE. Next, we described requirements representation and approaches, focusing on *i** 1.0, iStar 2.0, and use cases. We then introduced software quality, with emphasis on usability, appropriateness recognisability and learnability. We have seen that there are several methods for requirements' quality evaluation, such as metrics and the GQM approach; eye-tracking devices; EEG and EDA scanners; the NASA-TLX method, to perform subjective workload and cognitive load assessment; and GenderMag, to characterise stakeholders according to five facets. The combination of all these techniques can give us a multi-perspective about the models and the way stakeholders interact with them, enabling a richer analysis on the usability of the models and the problems they may have.

Chapter 3

QualitEva: A Mixed-Method Process for Quality Evaluation of Requirements Models

In this Chapter, we propose a mixed-method process for the quality evaluation of requirements models. The objective is to provide a step-by-step guide on how to perform (quasi-) experiments involving human subjects, and with the usage of (bio)metrics. True experiments have full randomization, and are difficult to perform in software engineering. As such, quasi-experiments are used instead, that is, experiment in which it has not been possible to assign participants in the experiments to groups randomly. Our process is based on Wohlin et al. guidelines [245] for experimentation in Software Engineering. However, we adapted those guidelines to the particular case of using (quasi-)experiments to evaluate the quality of requirements models through both the analysis of the models themselves, and the exploitation of human factors on how different people interact with them. We took into account our own knowledge, acquired after building and conducting several of these experiments (see, for example, [185–187]). We named it QualitEva (from ***Quality Eva**luation*). The activities carried out during the process are described using UML 2.0 activity diagrams.

3.1 Overview of the QualitEva Process

A (quasi-)experiment is a formal, rigorous and controlled investigation. As such, having a systematic process helps in the overall definition and execution of the experiment, in addition to reducing mistakes or erroneous conclusions due to a ill-defined evaluation [116, 225]. Furthermore, a process can be broken into clear and repeatable steps, which can be followed by both a senior or a novice experimenter. In Figure 3.1 we present an overview of the QualitEva process, based on [245].

The process is divided into the following 6 (six) main activities:

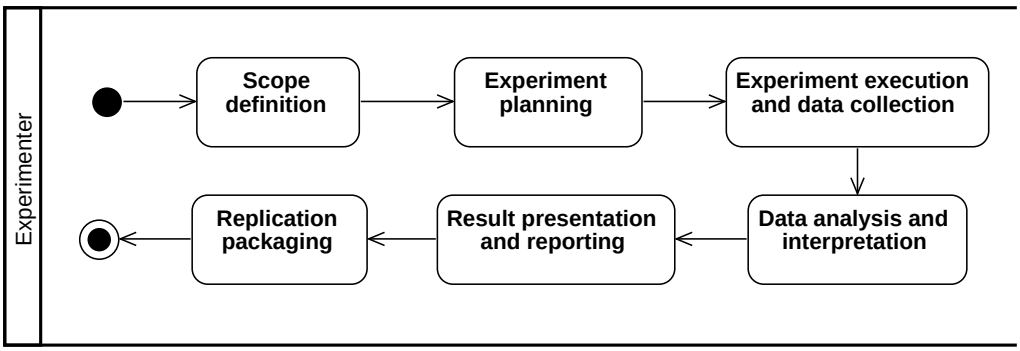


Figure 3.1: Overview of the QualitEva process, based on [245].

- **Scope definition** is related with the characterisation of the experiment in terms of problem to analyse, main objectives, goals, and high-level hypotheses.
- **Experiment planning** is related with the definition of the design, instrumentation, and threats to validity of the experiment.
- **Experiment execution and data collection** is related with the actual execution of the experiment, based on the planning, and the collection of data to be analysed in the next step.
- **Data analysis** is related with the evaluation of the obtained data.
- **Results presentation and reporting** is related with the documentation of the results, which can be published as a research paper, or as part of a dissertation.
- **Replication packaging** is related with the creation of a package for replication purposes, either by the same experimenter with a different group of participants, or by an independent researcher to further validate (or refute) the results.

All these steps are described in more detail in the next Sections.

3.2 Scope Definition

From a process point of view, the first step is to clearly scope the experiment. In Figure 3.2, we present the steps involved in this definition.

The first step is to precisely state the research problem the experimenter is trying to address, by defining a **problem statement**. Afterwards, the goals of the experiment must be defined. These goals are formulated based on the problem to be solved. We propose the GQM research goal template [7, 8], for their definition. However, before the goals can be defined, the experimenter needs to characterise: *(i)* the object of the study

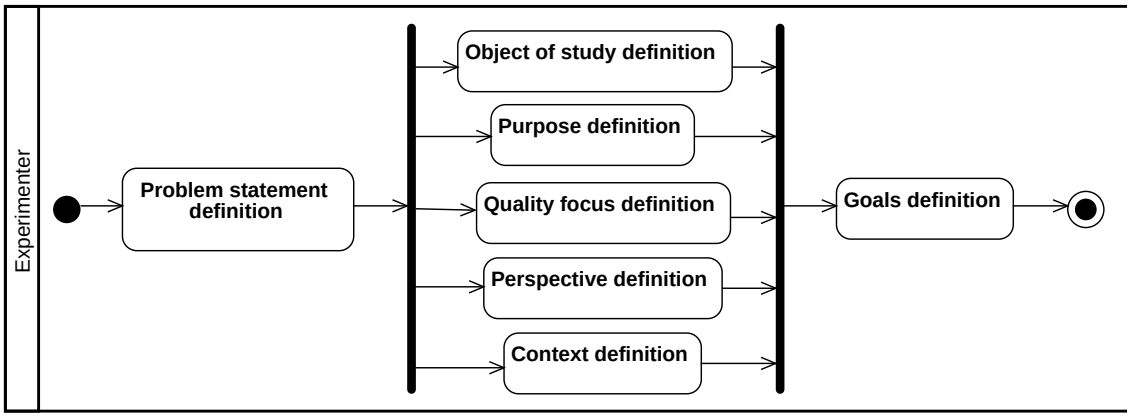


Figure 3.2: Experiment scope definition.

(what is studied); *(ii)* the purpose (what is the intention); *(iii)* the quality focus (which effect is studied); *(iv)* the perspective (whose view); and *(v)* context (where is the study conducted).

The **object of study** is the entity that is studied in the experiment. This is the requirements model the researcher wants to study. Some examples are *i**, KAOS, BPMN, or UML diagrams. The **purpose** defines the intention of the experiment. It may be, for example, to evaluate the impact of two different models, or two different layouts. The **quality focus** is the main effect under study in the experiment. We recommend analysing the quality characteristics available in the ISO/IEC 25023:2016 [107] (the latest version at the time of this writing, but always refer to the most recent version available). The **perspective** is the viewpoint from which the experiment results are interpreted, such as the researcher or experimenter. Finally, the **context** briefly defines which subjects and artefacts (in this case, requirements models) are used in the experiment.

With all these components determined, the experimenter is them able to properly define the goals with the GQM research goal template, as follows:

Scope: Template for the definition of the experiment's goals

Analyse <*object of the study*>
for the purpose of <*purpose*>
with respect to their <*quality focus*>
in the context of <*context*>.

By providing these informations, the experimenter can clearly state why (s)he is conducting the experiment. Furthermore, when the goals are defined, the experiment boundaries are delimited, and the experimenter has the foundations for the next step, which is planning the experiment.

3.3 Experiment Planning

After the definition of the scope of the experiment, the planning takes place. While the scope defined the **why**, the planning defines the **how**. This is a particularly critical step to ensure that the results from the experiment become useful. A poorly planned experiment can not be effectively controlled and the results may be compromised.

In Figure 3.3, we present the planning phase of the experiment, which can be divided into 10 (ten) steps. Based on the scope definition, presented in Section 3.2, the **context** definition selects the environment in which the experiment will take place. Then, the **hypotheses** are stated formally, each including a set of null and an alternative hypotheses. The next step is to determine the set of independent and dependent **variables** that will be used to evaluate the hypotheses, and to identify the **subjects** of the study. Afterwards, a suitable **experimental design** is chosen. Later on, the **collection process** is selected, as well as the **instrumentation** of the experiment. It is also important to consider the **validity** of the results that can be expected. This evaluation can force a change in the experimental design. Finally, a **pilot test** is executed, to validate the previously defined collection process and instrumentation. The pilot may show problems with the previously defined setup and cause those steps to change.

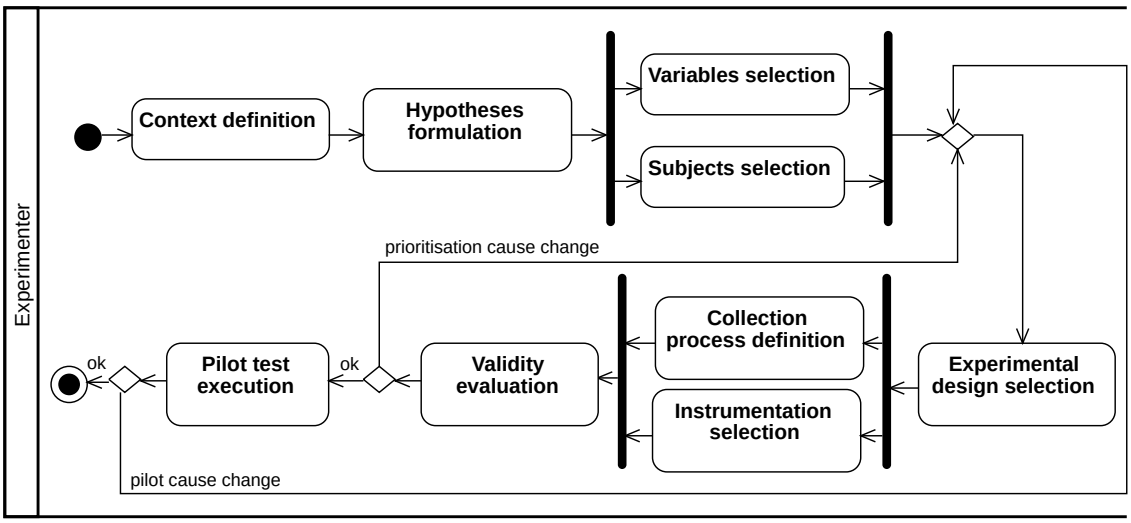


Figure 3.3: Experiment planning definition.

3.3.1 Context Definition

The context of the experiment can be characterised according to 4 (four) dimensions: *(i)* online *versus* offline; *(ii)* students *versus* professionals; *(iii)* toy examples *versus* real problems; and *(iv)* specific *versus* general.

The experiment can be conducted either **online** (*in vivo*) or **offline** (*in vitro*). The former is carried out in a professional environment, in a "real-world" scenario. However,

this setup has some risks, since the experiment may become intrusive to the daily work of the organisation, and even cause delays of their projects. As such, a common alternative is to perform the experiment offline, that is, not in a real-world environment. Two other experiment classifications were also proposed, the *in virtuo* and the *in silico* [230]. In the former, the experiments involve the interaction among participants and a computerised model of reality. In the latter, both the subjects and real world being described are computer models. The main advantages of *in virtuo* and *in silico* experiments are related with its lower cost and feasibility of replicating a real-world situation. However, in both cases, there is risk of not correctly representing the real-world elements' behaviour.

The subjects may be professional **practitioners**, **students**, or a combination of both. The first option produces results that are easily comparable to others obtained in a professional environment. However, using students as surrogates is less expensive but can make the results harder to extrapolate to the industrial reality. Nonetheless, students are often used as surrogates in Software Engineering experiments [63, 207]. Some studies have shown that graduate students can be a valid option in those experiments, especially when compared with novice practitioners [79, 98].

The type of problems to be used in the experiment must also be taken into consideration, in particular, the usage of **toy** or **real** problems. The resources available, including time the participants can devote to the experiment, or limitation in the equipment used to collect data from the participants, may justify the choice for a toy problem. Nonetheless, the results obtained from those experiments may not be generalisable to real problems. Toy problems can be used in an initial and exploratory study. If the results are satisfactory, the researcher may decide to scale up the experiment and use real problems.

Finally, an experiment can also be **specific** or **general**. In the former, the results are applicable to a niche, while in the latter they can be applicable to a larger population.

3.3.2 Hypotheses Formulation

The basis for the statistical analysis of an experiment is hypothesis testing. As such, hypotheses must first be formulated when planning the experiment, so that they can be tested with the results obtained afterwards. Two hypotheses have to be formulated, the **null** hypothesis (H_{0ij}) and the **alternative** hypothesis (H_{1ij}). In both cases, i stands for the goal identifier, while j corresponds to an hypothesis counter, if there are more than one hypothesis being tested for the same goal.

- **Null hypothesis (H_{0ij})** states that there are no real underlying trends or patterns in the experiment setting; the only reasons for differences in the observations are coincidental. This is the hypothesis that the experimenter wants to reject with as high significance as possible.

- **Alternative hypothesis (H_{1ij})** is the hypothesis in favour of which the null hypothesis is rejected. If the null hypothesis cannot be rejected, the alternative cannot be

accepted as well.

Hypothesis testing assumes a given level of significance, denoted by α, which represents a fixed probability of wrongly rejecting the null hypothesis, if it is true. The probability value, or ***p*-value**, of a statistical hypothesis test is the probability of getting a value equal to, or more extreme than that observed by chance alone, if the null hypothesis is true. Testing hypotheses involves different types of risks, such as the test rejecting a true hypothesis, or the test does not rejecting a false hypothesis. These risks are referred to as **type I error** and **type II error**. The **power** of a statistical test is the probability that the test will reveal a true pattern if the null hypothesis is false. An experimenter should choose a test with as high power as possible.

3.3.3 Variables selection

When selecting the appropriate variables, it is important to follow a goal-driven approach. This allows the information collected to be linked with the research goals. Furthermore, that way it is possible to prevent the collection of useless data for that experiment. The GQM approach [8], presented in Subsection 2.4.1, is used as a standard by the Empirical Software Engineering community.

There are 2 (two) types of variables to be defined: independent and dependent. The **independent** variables are the ones we can control and change in the experiment. For example, the modelling languages used, or the layout of a requirements model. These variables must have some effect on the dependent variables, and must be controlled. The **dependent** variables measure the effect of the independent variables. When using biometrics, the dependent variables are related with the metrics we want to measure and the biometric data we want to collect from the devices. When choosing the dependent variables, the measurement scale and the range of the variables must also be determined.

3.3.4 Subject Selection

The selection of the subjects is highly related with the generalisation of the results. In (quasi-)experiments involving human subjects, the subjects are also called participants. In order to generalise the results to the desired population, the selection must be representative for that population. The selection of subjects is also named sample from a population, or sampling. The sampling of a population can be either a **probability** or a **non-probability** sample. In the former, the probability of selecting each subject is known, while in the latter it is unknown. There are several sampling techniques [51]. In Empirical Software Engineering research, the most common is the non-probability convenience sampling, where the nearest and most convenient persons are selected as subjects.

The **size** of the sample also impacts the results when generalising. The larger the sample is, the lower the error becomes when generalising the results. When planning the experiment, the experimenter also has to determine the sample size needed to ensure

an adequate power level [115, 122]. This means deciding the standardised effect size in terms of Cohen's large, medium and small criteria [43], and use a power analysis to determine what sample size is necessary to achieve a power level of 0.8 (that is, 80% chance of detecting an effect if one genuinely exists). Given sufficient power, a non-significant effect is much more likely to indicate a negligible effect size, and a significant effect more likely to provide an unbiased estimate of the true effect size.

3.3.5 Experimental design selection

The selection of an experimental design is affected by the previous hypotheses formulation and variables selection. The design, in turn, influences the statistical approaches that can be followed when analysing the results from the experiment.

There are 3 (three) general design principles: randomisation, blocking, and balancing. **Randomisation** is used to average out an effect that might otherwise influence the test, by ensuring that observations are being made on independent random variables. On the other hand, **blocking** is used when there is an effect that may have an influence on the test, but is not the effect under evaluation. The effect is blocked by creating different groups within the sample. Finally, **balancing** is used to ensure that each group has an equal number of subjects. This is useful for improving the soundness of the statistical analysis. For a complete explanation of the different designs, see Wohlin et al. guidelines [245].

3.3.6 Collection process definition

When defining the collection process, it is important to consider not only who will conduct the experiment and collect the experimental data, as well as when, where and how the data collection will take place. If more than one person is going to run the experiment, all of them need to have a common understanding of the protocol to follow and the guidelines for the instructions to give to the participants. The goal is to minimise the data collection effort and variability, in order to ensure that data are collected in a consistent way throughout the entire process.

3.3.7 Instrumentation selection

The instrumentation process involves defining the artefacts and materials that will be used in the experiment. This also includes the choice of the biometrics devices and the development of tools (if needed) to support the measurements collected during the (quasi-)experiment. In Figure 3.4, we present the steps for creating these materials.

The instrumentation can be divided into 4 (four) main activities, which do not have a particular order to be performed. The experimenter needs to select the **problem domain** to which the requirements model should be applied to. Then, the models to be evaluated can be selected from a set of existing ones, or new models can be created, depending on the goal of the experiment. Afterwards, the complete description of the task the participants

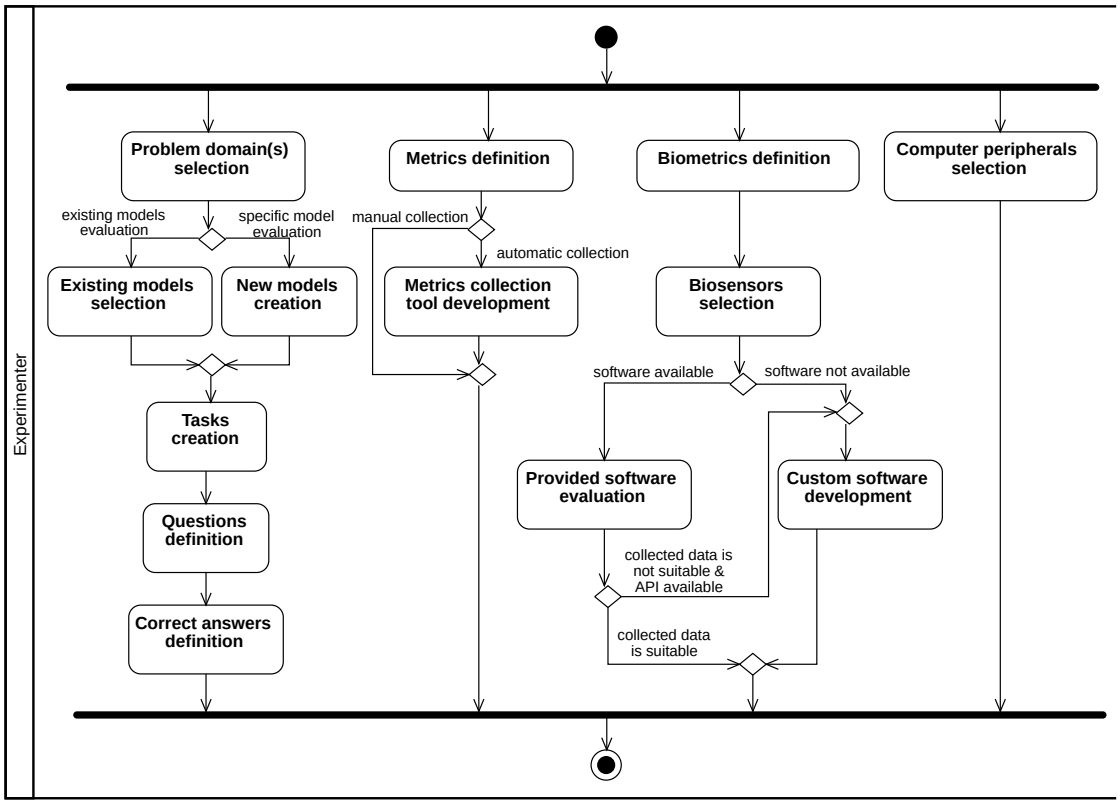


Figure 3.4: Experiment planning instrumentation.

will have to perform is created, questions are defined, and the correct answers (what the participants are suppose to reply or perform) are defined.

The definition of **metrics** can be achieved by applying the GQM approach. Those metrics can be collected manually or automatically. Manually collecting metrics can be time-consuming and error-prone but, on the other hand, the implementation of a tool to automatically collect them may not be feasible.

The definition of **biometrics** can also be achieved by applying the GQM approach. However, the devices that are going to be used need to be selected. As seen in Chapter 2, there are different types of devices, with a wide range of prices and functionalities. Some of them will have a collection and analysis software. Yet, with some low cost devices, an API is available, but the software needed to collect and analyse the data must be developed by the experimenter. It may also be the case where the software is available, but does not cover the experimenter needs. In that case, and if an API is available, the experimenter can develop a custom version of the software.

Finally, it is important to select the **computer peripherals** that are going to be used. Some eye-trackers, for example, have limitations in terms os screen size. Furthermore, depending on the task, the participants may need to have access to mouse and/or a keyboard. All this needs to be taken into account when planning the experiment.

3.3.8 Validity evaluation

An essential question concerning the results of an experiment is how valid those results actually are. An adequate validity means that the results obtained are valid for the population of interest. The results should then be valid for the population from which the sample is selected, and should also be generalisable to a broader population.

Validity threats can be divided into 4 (four) major classes: internal, external, construct, and conclusion validity [45]. The **internal validity** is concerned with influences that can affect the independent variables with respect to causality, without the experiment's knowledge or control. The external validity, on the other hand, is related with the generalisability of the results to outside the scope of the experiment. The **construct validity** is concerned with the relationship between theory and observation. Finally, the **conclusion validity** is related with issues that affect the ability to draw the correct conclusion about the relationships between the treatment and the results of the experiment.

There are conflicts between some of the types of the validity threats. Typically, when increasing one type, another may decrease. In different experiment, distinct types of validity can be prioritised in a different manner. As such, the clear definition of this prioritisation can cause the experimental design to change, as well as the collection process and the instrumentation selection.

3.3.9 Pilot study execution

After all the planning and instrumentation, it is necessary to guarantee that the experiment will be performed as expected. Conducting a **pilot study** is paramount to understand if anything needs to be changed, from the laboratorial or room settings, to the experimental materials used. A pilot study should be executed exactly as an normal experimental session would. However, the experimenter needs to be attentive to the way people interact with the materials, and the laboratory or room environment.

With eye-tracking devices, performing a pilot study is particularly useful to understand the amount of light a room must have, the correct position of the screen in relation with the window (if any), and the type of chair the participants should be sitting on. In the particular case of the chair, performing the pilot allowed us to realise that a chair with wheels would compromise the results, as the pilot participant tended to move the chair around, which caused the eye-tracker to become uncalibrated. Similar observations can be made with regard to other biometric devices, just as the usage of bracelets or watches when the devices needs to be placed on the wrist; or hair preparation, when the device needs to be placed in the head. Regarding the experimental materials, they may also be changed after the pilot study. It may be the case where the materials are too small, or the order of treatments is impacting the results (when that was not the goal of the study).

The results achieved by the participants in this pilot study cannot be used in the data evaluation.

3.4 Experiment Execution and Data Collection

After the experiment has been designed and planned, it must be executed and data have to be collected. In Figure 3.5, we present the steps involved in this process.

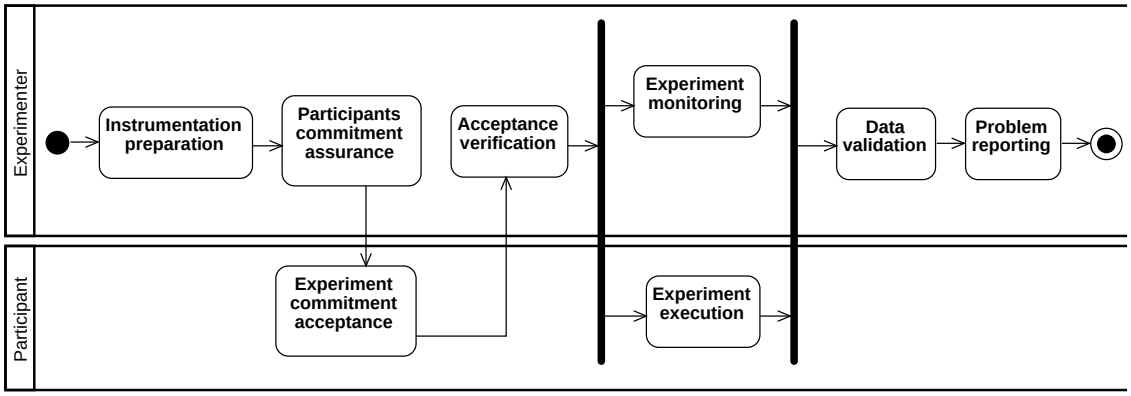


Figure 3.5: Experiment execution and data collection.

Before the experiment is actually executed, some **instrumentation preparation** has to be made. All experiment instruments, determined by the design of the experiment, and the methods that will be used for data collection, must be ready. The computer peripherals, if needed, have to be placed in the right location. The materials that are going to be used by the participants need to be ready and available when required.

When the participants do arrive, the experimenter has to assure the **participant commitment** to the experiment. This includes collecting clearance through a consent form, and informing the participants about the tasks. When biometrics devices are used, they should be explained to the participants, so that the usage of an unknown device does not cause any interference with the experiment. Furthermore, it is important to describe how the result of the experiment will be used and published, and what is being done to guarantee the anonymity of those results. In addition, it should be made clear to the participants that they are free to withdraw from the experiment at any moment. Any questions the participant may have, should be discussed before the experiment starts. After the participant accepts the **experiment commitment**, the experimenter **verifies the acceptance** and the experiment can start.

The **experiment execution** corresponds to the *actual* experiment, where participants perform the proposed tasks, and the relevant data are collected. The experimenter must ensure that the experiment is conducted according to the plan and design, by performing a **experiment monitoring**. Any problems detected should be registered for further analysis. An example is subject's mortality, which happens when a prospective participant does not participant, or a participant decides to quit the experiment.

After the experiment is executed, the experimenter needs to perform a **data validation**. The objective of this process is to ensure that the data has been correctly collected. Problems with the collection of data can occur if, for example, the collection tools

malfunction. The quality of the data is paramount, so that adequate conclusions and inferences can be made.

Finally, it is important to **report the problems**, as well as any deviation from the original plan and design. A complete and thorough identification of problems and how they were solved, has the purpose of helping with future replications of the experiment, as well as the identification of other potential validity threats.

3.5 Data Analysis

After the experiment has been executed and the data have been collected, the analysis can begin. In Figure 3.5, we present the steps involved in this process.

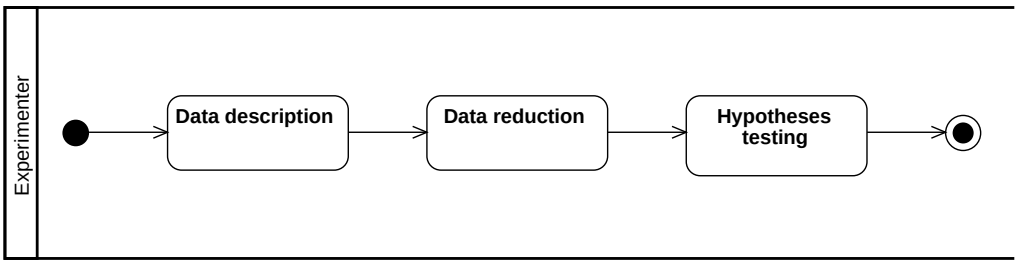


Figure 3.6: Experiment data analysis.

The first step is to perform a **data description**, where the data are characterised using descriptive statistics, dealing with the presentation and numerical processing of the data set. The goal of descriptive statistics is to show how the data set is distributed. Depending on the variables and scale types, the analysis changes. However, it is common to collect measures of central tendency, dispersion, and dependency.

After describing the data, the results allow the detection of errors in the data set, which may cause a **data reduction**. These errors can occur as systematic, or as outliers. In the former, those atypical cases can be incorrect data, which should be excluded from the analysis. As for the latter, outliers can give useful insights into the variables. Effective ways to identify outliers is to draw scatter plots, Q-Q plots, or box plots.

Finally, the **hypotheses testing** is performed. This will assess the hypotheses previously defined in the experiment plan. However, in order to select the correct statistical test to apply, the experimenter needs to check first if the pre-conditions for the test are satisfied. This includes understanding the distribution of the data in terms of normality, as well as variances among groups, if they are more than one. Some statistical test are robust to deviations from the normal distribution, different sample sizes, and variance in the samples, while others are not. For example, parametric tests are based on a model that involves a specific distribution. As such, in most cases, it is assumed that some of the parameters are normally distributed. On the other hand, non-parametric tests do not make the same type of assumptions concerning the distribution. However, the power of

parametric tests is generally higher, and requires fewer data. When selecting the test, it is therefore important to consider its applicability and power.

Further information on descriptive statistics and hypotheses testing, as well as common mistakes to avoid, is available in [121, 122, 245]. For performing the statistical analysis, there are several software programs available. We have been using IBM SPSS Statistics [103], which has a comprehensive set of statistical tools. However, it is proprietary software, with a subscription business model. There are some open source alternatives, like GNU PSPP [74] and JASP [112], for those who prefer a graphical interface. If the experimenter is at ease with programming languages, R [224] and Python [174] are also options to consider. A complete comparison of statistical software is available in [188].

3.6 Results Presentation and Reporting

Once the experiment and data analysis are finished, the intention is often to present the findings. This can be done, for example, in a research paper for a conference or journal, a report for decision-makers inside an organisation, or as educational material. In Figure 3.7, we present the steps involved in this process.

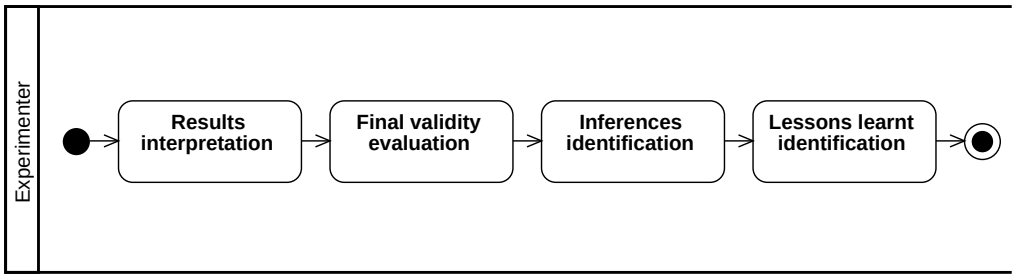


Figure 3.7: Experiment results presentation and reporting.

After the hypotheses testing is concluded, the experimenter has to perform the **results interpretation**. When the tests do not reject the null hypotheses, it is important to understand and identify the possible causes. This analysis enables the refinement of the theory, or the creation of a new rationale that explains the results.

Although a validity evaluation was made during the planning of the experiment, the execution may have introduced other threats to the validity. As such, it is important to reflect on the entire process and perform a **final validity evaluation**. The goal is not to diminish the value of the experimental work, but to identify opportunities for complementary studies.

Then, the **inferences identification** is made with reference to the validity. As such, factors that might have had an impact on the results should be described, and the experimenter have to estimate how the results obtained can be generalisable to beyond the experiment's sample.

Finally, the **lessons learnt identification** is an important step, as it can be useful for other experimenters and researchers, who have interested in replicating the experiment or perform a similar one.

For documenting the experimental process, as well as the results interpretation, inferences and lessons learnt, we recommend following Jedlitschka et al. guidelines [113] on how to report (quasi-)experiments in Software Engineering. Depending on the requirements of the publisher, it may be necessary to adapt the length of the final report. However, following a well-established document structure, with the sufficient amount of details, makes it easier for independent researchers to find the relevant information and to perform replications.

3.7 Replication Packaging

The general function of replications is to verify the results of an experiment. When an experiment is not replicated, it is not possible to distinguish whether the results were produced by chance, and the events occurred accidentally; the results were artifactual, and the events occurred due to the experimental configuration but do not exist in reality; or the results actually conform to a pattern existing in reality [77]. Different replication types help to clarify which of these types of results an experiment has. As such, replications serve several different and more specific purposes, depending on the changes that are introduced [189].

There are 6 (six) verification functions for Software Engineering experiment replications [77]: *(i)* control sampling error, *(ii)* control protocol independence, *(iii)* understand operationalisation limits, *(iv)* understand populations limits, *(v)* control experimenters independence, and *(vi)* validate hypotheses.

Although the number of internal replications (where the original researchers performed the replication) is growing, the number of external replications (where independent researchers performed the replication) is not growing as much. This indicates that the research community is more aware of the importance of performing replications, but replicating the experiments of other researchers is not an established practice [205]. One of the reasons for this lack of external replications, is that the detailed information about the original experiment is not fully available. A way to mitigate this is to have a repository to store all the relevant data about the experiments [205]. Some examples of these repository can be explored in [173, 251].

The reporting of the results, when well detailed and described, can be used to perform independent replications of the experiment. However, in the particular cases of controlling both sampling errors and experimenters independence, the operationalisation, protocol and population should not be changed. These types of replications are useful to understand the natural variation of results and if an event was not due to type I errors, addressing the conclusion validity threats. However, in order to properly perform

these types of replications, it is paramount that the original materials are available. As such, we argue it is also important to provide a replication packaging.

The experimental materials used in the original experiment can be offered in technical report, or on a webpage from where the materials can be downloaded. Either way, the experimental report, described in Subsection 3.6, should also be made available.

3.8 Summary

We presented the QualitEva process for the quality evaluation of requirements models, based on Wohlin et al. guidelines [245] for experimentation in Software Engineering. The process has 6 (six) main activities: scope definition, experiment planning, experiment execution and data collection, data analysis, results presentation and reporting, and replication packaging. We describe each activity in detail. As such, this Chapter can be used as a step-by-step guide to plan and execute (quasi-)experiments involving human subjects, and with the usage of (bio)metrics.

Chapter 4

(Bio)Metrics for the Evaluation of Requirements Models

In this Chapter, we propose a set of (bio)metrics for the evaluation of requirements models. The definition of these (bio)metrics was performed following the GQM approach (presented in Subsection 2.4.1). Our goals are related with the evaluation of the *(i)* accuracy achieved by stakeholders when performing tasks on requirements models, as well as their *(ii)* speed *(iii)* visual ease; *(iv)* mental ease; *(v)* emotional ease; and *(vi)* perceived effort while performing those tasks. We further propose metrics for the evaluation of *(vii)* *i** models. All these (bio)metrics can give us information about the models themselves, as well as the way different stakeholder interact with them. In the metrics related with *i** models, we are interested in analysing their complexity and completeness. As such, they can be perceived as subset of accuracy metrics. However, we decided to separate them in this Chapter for readability purposes.

4.1 Accuracy Metrics

In Table 4.1 we summarise the result of applying the GQM approach to identify a set of metrics that allows satisfying the goal of **accuracy evaluation**. The first column (*Question*) presents questions that allow evaluating whether the overall goal is being achieved. The second column (*Metric*) shows metrics that provide quantitative information to answer the corresponding question.

Question Q1 is concerned with the quality of the answer given by a stakeholder to a question related with a requirements model, that is, if the answer had substantially more relevant results than irrelevant ones. Question Q2 is targeted at quantity, that is, if an answer has most of the relevant results, even if it has "noise" (several irrelevant elements). Finally, question Q3 quantities the overall accuracy achieved by the stakeholder.

Table 4.1: Goal-Question-Metric for the evaluation of accuracy.

Goal: Evaluate the accuracy of stakeholders when performing tasks on requirements models	
Question	**Metric**
Q1 - How can we measure the exactness of an answer given by stakeholder on a particular task?	**M1** – Precision
Q2 – How can we measure the completeness of an answer given by a stakeholder on a particular task?	**M2** – Recall
Q3 – How can we measure the overall accuracy of an answer given by a stakeholder on a particular task?	**M3** – F-measure

For each metric, we provide a Table containing *(i)* an informal definition, in natural language; *(ii)* a formal definition using a mathematical expression; and *(iii)* an example of application.

Regarding question Q1 (Table 4.2), the value of *precision* (also known as positive predictive level) is measured by the fraction of relevant elements among the ones retrieved by the stakeholder. It answers the question *"How many retrieved elements are relevant?"*. Precision reaches its best value at 1 (perfect precision) and worst at 0.

Table 4.2: **Q1** – How can we measure the exactness or quality of an answer given by stakeholder on a particular task?

Metric	Precision
Informal definition	Fraction of elements retrieved by stakeholders which are relevant.
Formal definition	$\frac{number\ of\ relevant\ elements\ retrieved}{total\ number\ of\ retrieved\ elements}$
Example	In a creation task, a stakeholder has to create 3 model elements. If (s)he adds 1 relevant element and 1 irrelevant element, the precision is $\frac{1}{2}$.

Concerning question Q2 (Table 4.3), the value of *recall* (also known as sensitivity) is measured by the fraction of relevant elements that have been retrieved by the stakeholder among the total number of relevant elements. It answers the question *"How many relevant elements are retrieved?"*. Recall reaches its best value at 1 (perfect recall) and worst at 0.

Table 4.3: **Q2** – How can we measure the completeness of an answer given by a stakeholder on a particular task?

Metric	Recall
Informal definition	Fraction of relevant elements retrieved by stakeholders among the total number of elements.
Formal definition	$\frac{number\ of\ relevant\ elements\ retrieved}{total\ number\ of\ relevant\ elements}$
Example	In a creation task, a stakeholder has to create 3 model elements. If (s)he adds 1 relevant element and 1 irrelevant element, the recall is $\frac{1}{3}$.

With respect to question Q3 (Table 4.4), the value of f-measure (also known as balanced f-score or F_1-score) is measured by a combination of precision and recall, computing an harmonic mean. F-measure reaches its best value at 1 (perfect precision and recall) and worst at 0.

Table 4.4: **Q3** – How can we measure the overall accuracy of an answer given by a stakeholder on a particular task?

Metric	F-measure
Informal definition	Combination of precision and recall, providing an harmonic mean.
Formal definition	$\frac{2 * (precision * recall)}{(precision + recall)}$
Example	In a creation task, a stakeholder has to create 3 model elements. If (s)he adds 1 relevant element and 1 irrelevant elements, the precision is $\frac{2}{5}$.

4.2 *i** Models Metrics

In a previous work, we defined metrics, formalised in OCL, about the complexity and completeness of *i** 1.0 models [81, 82]. In this dissertation, we adapted those metrics for iStar 2.0. For the sake of brevity, we only present those in this Chapter. The goal is to evaluate: *(i)* the complexity and *(ii)* the completeness of iStar 2.0 models.

4.2.1 Introduction to the *i** Metrics Set

In Tables 4.5 and 4.6, we summarise the result of applying the GQM approach to propose a set of metrics that allows satisfying the goals of complexity and completeness evaluation of iStar 2.0 models. The first column (*Question*) presents questions that allow evaluating whether the overall goal is being achieved. The second column (*Metric*) shows a set of metrics that provide quantitative information to answer the corresponding question.

The **complexity evaluation** goal, in Table 4.5, is related with the model and its elements. Question Q1 concerns complexity, as perceived when regarding the model as a whole. In particular, we are interested in the number of actors and in the number of elements, within a given model. The next set of questions are targeted to assessing the complexity of model elements, namely the amount of responsibilities supported by an actor in the model (Q2), and the number of decompositions of actor's goals (Q3), qualities (Q4) and tasks (Q5). For each of these elements-centred questions, we define a basic metric and 3 (three) additional distribution metrics presenting the minimum, maximum and average values for the basic metric. Questions Q6 and Q7 quantify the dependency relationships of an actor, and we are interested in the percentage of outgoing (Q6) and incoming dependencies (Q7) of such actor. Lastly, Q8 allows to infer if the complexity of a certain actor is related with its type (that is, actor, agent, or role).

The **completeness evaluation** goal, in Table 4.6, is related with the requirements identified. The first questions are targeted to assessing the detail of actors' specification, and

Table 4.5: Goal-Question-Metric for evaluating the complexity of iStar 2.0 models.

Goal: Evaluate the complexity of iStar 2.0 models	
Question	**Metric**
Q1 – How complex is the model, concerning the actors and elements?	**M1** – Number of actors **M2** – Number of elements
Q2 – Does an actor have too much responsibility in the model?	**M3** – Number of elements of an actor **M4** – Minimum number of elements of an actor **M5** – Maximum number of elements of an actor **M6** – Average number of elements of an actor
Q3 – How complex is an actor's goal, with respect to its decompositions?	**M7** – Number of refinements of a goal **M8** – Minimum number of refinements of a goal **M9** – Maximum number of refinements of a goal **M10** – Average number of refinements of a goal
Q4 – How complex is an actor's quality, with respect to its decompositions?	**M11** – Number of contributions of a quality **M12** – Minimum number of contributions of a quality **M13** – Maximum number of contributions of a quality **M14** – Average number of contributions of a quality
Q5 – How complex is an actor's task, with respect to its decompositions?	**M15** – Number of refinements of a task **M16** – Minimum number of refinements of a task **M17** – Maximum number of refinements of a task **M18** – Average number of refinements of a task
Q6 – Is an actor too dependent in the model?	**M19** – Percentage of outgoing dependencies
Q7 – Does an actor have too many dependencies in the model?	**M20** – Percentage of incoming dependencies
Q8 – Is there a variation in the average complexity of the different types of actors?	**M21** – Number of elements inside an actor **M22** – Number of elements inside an agent **M23** – Number of elements inside a role

the detail level of goals and qualities. In particular, we are interested in the percentage of actors with a specific type (Q9), goals refinements or qualifications (Q10) and qualities with contributions and qualifications (Q11). Question Q12 quantifies the percentage of actors with elements inside its boundary. Finally, the last questions allows to assess how complete is the model and how close we are to finish the modelling process. In particular, we are interested in the assignment of responsibilities to an actor (Q13) and in the assignment of links, namely dependencies and associations, to and between actors (Q14).

4.2.2 *i** Metrics Definition

For each question defined in Subsection 4.2.1, we provide a Table containing information concerning *(i)* the symptom a requirements engineer should be alert to (in terms of detecting *"unusual values"*, when compared to other projects - more precisely, outlier and extreme values); *(ii)* the potential problem this symptom may indicate (*"suspicious"* metrics values do not necessarily imply that there is a problem, they just suggest it may

Table 4.6: Goal-Question-Metric for evaluating the completeness of iStar 2.0 models.

Goal: Evaluate the completeness of iStar 2.0 models	
Question	**Metric**
Q9 – How specific are the actors?	**M24** – Percentage of specific actors
Q10 – How detailed are the goals?	**M25** – Percentage of goals with refinements or qualifications
Q11 – How detailed are the qualities?	**M26** – Percentage of qualities with contributions or qualifications
Q12 – How detailed is the SR model with respect to its actors?	**M27** – Percentage of actors with elements inside
Q13 – How close are we to end the assignment of responsibilities to an actor?	**M28** – Percentage of actors without unconnected elements inside
Q14 – How close are we to end the assignment of links to the actors?	**M29** – Percentage of actors with dependencies or associations

be worth checking if there is one, thus helping in early problem identification and mitigation); *(iii)* a suggested action that the requirements engineer may want to take, if after inspecting the corresponding model elements (s)he decides there is an actual problem worth fixing; *(iv)* an informal definition of the metrics specified to answer it; and *(v)* a formal definition using OCL upon the meta-model fragment we present in Figure 4.1.

When required, we also include pre-conditions in the formal definition. For example, when defining metrics to compute the average decomposition of goals, qualities, or tasks, a typical pre-condition is to ensure that there are goals, qualities, or tasks, to be decomposed. Elements without decompositions may have been modelled in order to be final elements. It would not make sense analysing the extent to which they are decomposed. For the sake of brevity, we omit trivial auxiliary metrics definitions with basic counts. The auxiliary metrics (AM) can be found in Appendix A.

Regarding question Q1 (Table 4.7), the values of `NAct` (number of actors) and `NElem` (number of elements) are measures for the SD/SR model size. Size can be used as a surrogate for overall model complexity, and used to compare the complexity among different models. Different candidate models for the same system can be compared, using these metrics, with respect to their overall complexity. Work on other paradigms, such as Object-Oriented, has collected empirical evidence on the positive correlation between size and complexity [59, 218]. Over-simplistic models may be insufficiently detailed, leading to problems in their understandability. On the other hand, if the system is unnecessarily complex, understandability problems may also occur.

Concerning question Q2 (Table 4.8), a high value for `NEA` (number of elements of an actor) can be an indicator that a particular actor has too much responsibility in the model, being harder to manage the responsibilities in an efficient manner. The minimum, maximum and average values help the requirements engineer recognising cases where the responsibility is higher or lower than expected. Complexity can also be used for

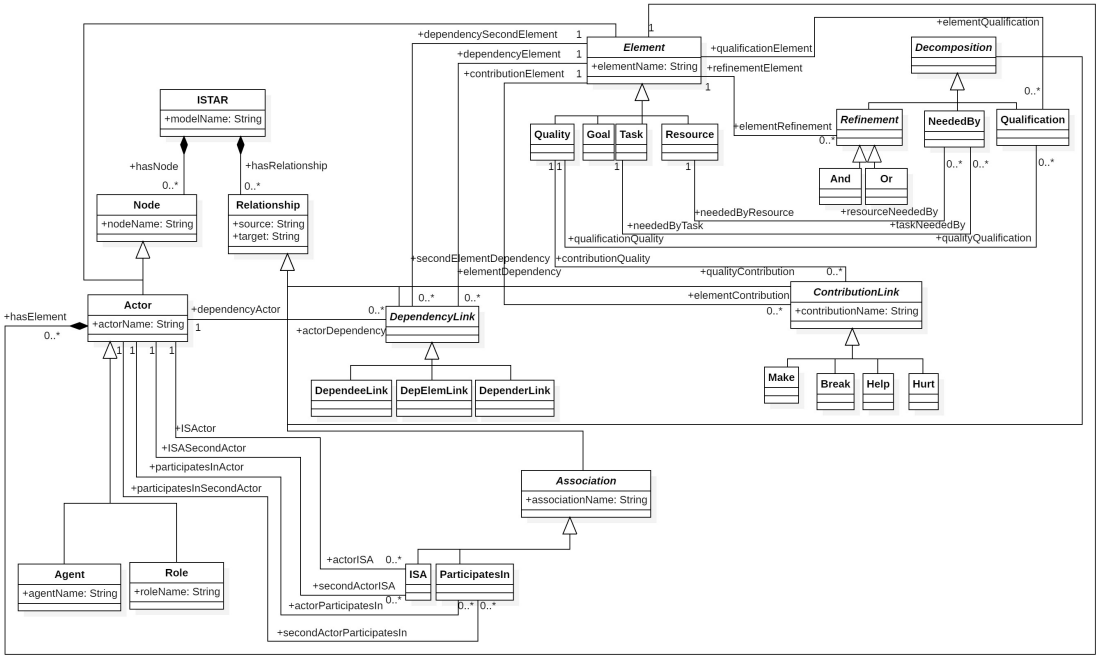


Figure 4.1: iStar 2.0 metamodel, adapted from [49].

Table 4.7: **Q1** – How complex is the model, concerning the actors and elements?

Symptom	The size of the model is unusually lower, or higher, than in most models.
Potential problem	The model may be over-simplistic, or unnecessarily complex, leading to problems in the understandability of the system.
Suggested action	Consider revising the model. If necessary, add more detail, or remove accidental complexity.
Metric	**NAct** – *Number of Actors*
Informal definition	Total number of actors in the SD/SR model

Formal definition	`context ISTAR`
	`def:NAct():Integer = self.hasNode ->`
	`select(n:Node | n.oclIsKindOf(Actor)) -> size()`

Metric	**NElem** – *Number of Elements*
Informal definition	Total number of elements in the SD/SR model
Formal definition	`context ISTAR` `def:NElem():Integer = self.NEOAB() + self.NEIAB()`
Requires	**NEOAB** – **N**umber of **E**lements **O**utside **A**ctors' **B**oundaries (AM A.1) **NEIAB** – **N**umber of **E**lements **I**nside **A**ctors' **B**oundaries (AM A.2)

supporting project estimation efforts.

Table 4.8: **Q2** – Does an actor have too many responsibilities in the model?

Symptom	The actor has an unusually high number of internal model elements.
Potential problem	The actor may have too many responsibilities in the model.
Suggested action	This actor may be a good candidate for further scrutiny. Consider decomposing this actor into several specialised sub-actors and distributing his responsibilities among them. If the system has no outliers, the assignment of responsibilities is probably well balanced.
Metric	**NEA** – ***N**umber of **E**lements of an **A**ctor*
Informal definition	Number of elements inside an actor's boundary in the SR model

Formal definition	`context Actor`
	`def:NEA():Integer = self.hasElement ->`
	` select(e:Element | e.oclIsKindOf(Element)) -> size()`

Metric	**MinNEA** – ***Min**imum **N**umber of **E**lements of an **A**ctor*
Informal definition	Minimum number of elements inside an actor's boundary in the SR model

Formal definition	`context ISTAR`
	`def:MinNEA(): Integer = self.hasNode ->`
	` select(n:Node | n.oclIsKindOf(Actor)) ->`
	`  iterate(n:Node; min:Integer = -1 |`
	`  let nea:Integer = n.oclAsType(Actor).NEA() in`
	`  if min = -1 then nea else min.min(nea) endif)`

Requires	**NEA** – **N**umber of **E**lements of an **A**ctor
Metric	**MaxNEA** – ***Max**imum **N**umber of **E**lements of an **A**ctor*
Informal definition	Maximum number of elements inside an actor's boundary in the SR model

Formal definition	`context ISTAR`
	`def:MaxNEA(): Integer = self.hasNode ->`
	` select (n:Node | n.oclIsKindOf(Actor)) ->`
	`  iterate (n:Node; max:Integer = -1 |`
	`  let nea:Integer = n.oclAsType(Actor).NEA() in`
	`  if max = -1 then aux else max.max(nea) endif)`

Requires	**NEA** – **N**umber of **E**lements of an **A**ctor
Metric	**AvgNEA** – ***Av**erage **N**umber of **E**lements of an **A**ctor*
Informal definition	Average number of elements inside an actor's boundary in the SR model

Formal definition	`context ISTAR::AvgNEA()`
	`pre:self.NAct() > 0`
	
	`context ISTAR`
	`def:AvgNEA(): Double = self.NEA() / self.NAct()`

Requires	**NEA** – **N**umber of **E**lements of an **A**ctor **NAct** – **N**umber of **Act**ors

Questions Q3, Q4 and Q5, (Tables 4.9, 4.10, and 4.11, respectively) provide different perspectives on the complexity associated with a particular actor. The value of NDG (number of decompositions of an actor's goal), presented in Q3, measures the complexity of the goal decompositions associated with an actor. The value of NDQ (number of decompositions of an actor's quality), presented in Q4, measures the complexity of the quality decompositions associated with an actor. Finally, the value of NDT (number of decompositions of an actor's task), presented in Q5, measures the complexity of the task

decompositions associated with an actor. The minimum, maximum and average values for NDG, NDQ and NDT help the requirements engineer identifying out of the ordinary goal, quality, or task decomposition complexities, respectively. The minimum value is computed only for goals, qualities, or tasks, which are decomposed. As such, it excludes leaf elements in its computation. Including too many design details in the model, by over-decomposing goals, qualities and tasks, may obfuscate the model, making it harder to understand.

Table 4.9: **Q3** – How complex is an actor's goal, with respect to its decompositions?

Symptom	An actor's goal has an unusually high number of decompositions.
Potential problem	The goal may be over-decomposed.
Suggested action	This goal may be a good candidate for further scrutiny. Consider abstracting out this goal, if it is over-decomposed. If the actor has no outlier goals, their decomposition is probably well balanced.
Metric	***NDG*** **–** ***N****umber of* ***D****ecompositions of a* ***G****oal*
Informal definition	Number of decompositions associated with a goal in the SR model

Formal definition	`context Goal`
	`def:NDG():Integer = self.elementRefinement ->`
	`select(re:Refinement | re.oclIsKindOf(Refinement)) -> size ()`

Metric	**MinNDG** **–** ***Min****imum* ***N****umber of* ***D****ecompositions of a* ***G****oal*
Informal definition	Minimum number of decompositions associated with a goal in the SR model

Formal definition	`context Actor`
	`def:MinNDG(): Integer = self.hasElement ->`
	`select (e:Element | e.oclIsKindOf(Goal) and`
	`e.oclAsType(Goal).NDG() > 0) ->`
	`iterate (e:Element; min:Integer = -1 |`
	`let ndg:Integer = e.oclAsType(Goal).NDG() in`
	`if min = -1 then ndg else min.min(ndg) endif)`

Requires	**NDG** – **N**umber of **D**ecompositions of a **G**oal
Metric	**MaxNDG** **–** ***Max****imum* ***N****umber of* ***D****ecompositions of a* ***G****oal*
Informal definition	Maximum number of decompositions associated with a goal in the SR model

Formal definition	`context Actor`
	`def:MaxNDG():Integer = self.hasElement ->`
	`select(e:Element | e.oclIsKindOf(Goal) and`
	`e.oclAsType(Goal).NDG () > 0) ->`
	`iterate (e:Element; max:Integer = -1 |`
	`let ndg:Integer = e.oclAsType(Goal).NDG() in`
	`if max = -1 then ndg else max.max(ndg) endif)`

Requires	**NDG** – **N**umber of **D**ecompositions of a **G**oal
Metric	**AvgNDG** **–** ***Av****erage* ***N****umber of* ***D****ecompositions of a* ***G****oal*
Informal definition	Average number of decompositions associated with a goal in the SR model

Formal definition	`context Actor::AvgNDG()`
	`pre:self.NGWDI() > 0`
	
	`context Actor`
	`def:AvgNDG(): Double = self.NDG() / self.NGWDI()`

Requires	**NDG** – **N**umber of **D**ecompositions of a **G**oal **NGWDI** – **N**umber of **G**oals **W**ith **D**ecompositions **I**nside (AM A.3)

Table 4.10: **Q4** – How complex is an actor's quality, with respect to its decompositions?

Symptom	An actor's quality has an unusually high number of decompositions.
Potential problem	The quality may be over-decomposed.
Suggested action	This quality may be a good candidate for further scrutiny. Consider abstracting out this quality, if it is over-decomposed. If the actor has no outlier qualities, their decomposition is probably well balanced.
Metric	**NDQ** – ***N**umber of **D**ecompositions of a **Q**uality*
Informal definition	Number of decompositions associated with a quality in the SR model

Formal definition	`context Quality`
	`def:NDQ(): Integer = self.qualityContribution ->`
	` select(cl:ContributionLink | cl.oclIsKindOf(ContributionLink))`
	`  -> size ()`

Metric	**MinNDQ** – ***Min**imum **N**umber of **D**ecompositions of a **Q**uality*
Informal definition	Minimum number of decompositions associated with a quality in the SR model

Formal definition	`context Actor`
	`def:MinNDQ():Integer = self.hasElement ->`
	` select(e:Element | e.oclIsKindOf(Quality) and`
	`  e.oclAsType(Quality).NDQ() > 0) ->`
	`   iterate(e:Element; min:Integer = -1 |`
	`    let ndq:Integer = e.oclAsType(Quality).NDQ() in`
	`     if min = -1 then ndq else min.min(ndq) endif)`

Requires	**NDQ** – **N**umber of **D**ecompositions of a **Q**uality
Metric	**MaxNDQ** – ***Max**imum **N**umber of **D**ecompositions of a **Q**uality*
Informal definition	Maximum number of decompositions associated with a quality in the SR model

Formal definition	`context Actor`
	`def:MaxNDQ(): Integer = self.hasElement ->`
	` select(e:Element | e.oclIsKindOf(Quality) and`
	`  e.oclAsType(Quality).NDQ () > 0) ->`
	`   iterate(e:Element; max:Integer = -1 |`
	`    let ndq:Integer = e.oclAsType(Quality).NDQ() in`
	`     if max = -1 then ndq else max.max(ndq) endif)`

Requires	**NDQ** – **N**umber of **D**ecompositions of a **Q**uality
Metric	**AvgNDS** – ***Avg**erage **N**umber of **D**ecompositions of a **Q**uality*
Informal definition	Average number of decompositions associated with a quality in the SR model

Formal definition	`context Actor::AvgNDQ ()`
	`pre:self.NQWDI () > 0`
	
	`context Actor`
	`def:AvgNDQ():Double = self.NDQ() / self.NQWDI()`

Requires	**NDQ** – **N**umber of **D**ecompositions of a **Q**uality **NQWDI** – **N**umber of **Q**ualities **W**ith **D**ecompositions **I**nside (AM A.4)

Concerning questions Q6 and Q7 (Tables 4.12 and 4.13, respectively), the values of POD (percentage of outgoing dependencies) and PID (percentage of incoming dependencies) are measures of an actor's dependency links in the SD/SR model. These values can be used to verify if an actor is a *source* (meaning the actor only has outgoing dependencies), or a *sink* (meaning the actor only has incoming dependencies), allowing the identification of pathological situations and actors' archetypes in the system. Furthermore, too many

Table 4.11: **Q5** – How complex is an actor's task, with respect to its decompositions?

Symptom	An actor's task has an unusually high number of decompositions.
Potential problem	The task may be over-decomposed.
Suggested action	This task may be a good candidate for further scrutiny. Consider abstracting out this task, if it is over-decomposed. If the actor has no outlier tasks, their decomposition is probably well balanced.
Metric	**NDT** – ***N**umber of **D**ecompositions of a **T**ask*
Informal definition	Number of decompositions associated with a task in the SR model

Formal definition	`context Task`
	`def:NDT():Integer = self.elementRefinement ->`
	` select(re:Refinement | re.oclIsKindOf(Refinement)) -> size ()`

Metric	**MinNDT** – ***Min**imum **N**umber of **D**ecompositions of a **T**ask*
Informal definition	Minimum number of decompositions associated with a task in the SR model

Formal definition	`context Actor`
	`def:MinNDT(): Integer = self.hasElement ->`
	` select (e:Element | e.oclIsKindOf(Task) and`
	`  e.oclAsType(Task).NDT() > 0) ->`
	`  iterate (e:Element; min:Integer = -1 |`
	`  let ndt:Integer = e.oclAsType(Task).NDT() in`
	`  if min = -1 then ndt else min.min(ndt) endif)`

Requires	**NDT** – **N**umber of **D**ecompositions of a **T**ask
Metric	**MaxNDT** – ***Max**imum **N**umber of **D**ecompositions of a **T**ask*
Informal definition	Maximum number of decompositions associated with a task in the SR model

Formal definition	`context Actor`
	`def:MaxNDT():Integer = self.hasElement ->`
	` select(e:Element | e.oclIsKindOf(Task) and`
	`  e.oclAsType(Task).NDT () > 0) ->`
	`  iterate (e:Element; max:Integer = -1 |`
	`  let ndt:Integer = e.oclAsType(Task).NDT() in`
	`  if max = -1 then ndt else max.max(ndt) endif)`

Requires	**NDT** – **N**umber of **D**ecompositions of a **T**ask
Metric	**AvgNDT** – ***Av**erage **N**umber of **D**ecompositions of a **T**ask*
Informal definition	Average number of decompositions associated with a task in the SR model

Formal definition	`context Actor::AvgNDT()`
	`pre:self.NTWDI() > 0`
	
	`context Actor`
	`def:AvgNDT(): Double = self.NDT() / self.NTWDI()`

Requires	**NDT** – **N**umber of **D**ecompositions of a **T**ask **NTWDI** – **N**umber of **T**asks **W**ith **D**ecompositions **I**nside (AM A.5)

dependencies increase the complexity and reduces the encapsulation and reuse potential of the actors. Excessive dependencies also limit the understandability and maintainability of the system. In addition, if too many actors depend on a particular actor, changes in this actor may have ripple effects through the other actors, potentially reducing the maintainability of the system.

Table 4.12: **Q6** – Is an actor too dependent in the model?

Symptom	An actor has an unusually high number of outgoing dependencies.
Potential problem	The actor may be too dependent on other actors to achieve its goals.
Suggested action	This actor may be a good candidate for further scrutiny. Consider balancing the number of outgoing and incoming dependencies among actors. If there are no outliers, the dependencies are probably well balanced.
Metric	**POD** – *Percentage of Outgoing Dependencies*
Informal definition	Percentage of outgoing dependencies of an actor in the SD/SR model

Formal definition	`context Actor::POD()`
	`pre:self.ND() > 0`
	
	`context Actor`
	`def:POD():Double = self.NOD() / self.ND()`

Requires	**NOD** – **N**umber of **O**utgoing **D**ependencies (AM A.6) **ND** – **N**umber of **D**ependencies (AM A.12)

Table 4.13: **Q7** – Does an actor have too many dependencies in the model?

Symptom	An actor has an unusually high number of incoming dependencies.
Potential problem	If too many actors depend on a particular actor, changes in this actor may have ripple effects through the other actors. This potentially reduces the maintainability of a system.
Suggested action	This actor may be a good candidate for further scrutiny. Consider balancing the number of outgoing and incoming dependencies among actors. If there are no outliers, the dependencies are probably well balanced.
Metric	**PID** – *Percentage of Incoming Dependencies*
Informal definition	Percentage of incoming dependencies of an actor in the SD/SR model

Formal definition	`context Actor::PID()`
	`pre:self.ND() > 0`
	
	`context Actor`
	`def:PID():Double = self.NID() / self.ND()`

Requires	**NID** – **N**umber of **I**ncoming **D**ependencies (AM A.13) **ND** – **N**umber of **D**ependencies (AM A.12)

Question Q8 (Table 4.14) provide information about the complexity of generic and specific actors, that is: actors, agents, and roles. The value of `NEIAct` (number of elements inside an actor), `NEIAg` (number of elements inside an agent), and `NEIR` (number of elements inside a role) allows to verify if the complexity of a certain actor is related with its type. It might be the case that a particular type of actor is frequently over- or under-specified, which would reflect on the typical complexity of actors of that type. Note that excessive use of the specific actor notations might lead to more complex models that

might become harder to deal with and understand.

Table 4.14: **Q8** – Is there a variation in the average complexity of the different types of actors?

Symptom	The actors from different types have a significantly different complexity. This is observable by statistically comparing the distributions of complexity by actor type.
Potential problem	It might be the case that a particular type of actor is frequently over- or under-specified, which would reflect on the typical complexity of actors of that type. On the other hand there might be a good reason for making a particular actor type more (or less) complex than the other ones.
Suggested action	Consider comparing the average complexity of the different actor types with the one found in other systems. If it is significantly different, further investigate whether this results from the essential complexity of the system, or from some accidental factor (such as over-, or under-specification of the actors).
Metric	**NEIAct** – ***N**umber of **E**lements **I**nside an **A**ctor*
Informal definition	Number of elements inside an actor's boundary in the SR model

Formal definition	`context ISTAR`
	`def:NEIAct():Integer = self.NEIAB() -`
	`(self.NEIAgB() + self.NEIRB())`

Requires	**NEIAB** – **N**umber of **E**lements **I**nside **A**ctors' **B**oundaries (AM A.2) **NEIAgB** – **N**umber of **E**lements **I**nside **Ag**ents' **B**oundaries (AM A.19) **NEIRB** – **N**umber of **E**lements **I**nside **R**oles' **B**oundaries (AM A.20)
Metric	**NEIAg** – ***N**umber of **E**lements **I**nside an **Ag**ent*
Informal definition	Number of elements inside an agent's boundary in the SR model

Formal definition	`context Agent`
	`def:NEIAg():Integer = self.hasElement ->`
	`select(e:Element | e.oclIsKindOf(Element)) -> size()`

Metric	**NEIR** – ***N**umber of **E**lements **I**nside a **R**ole*
Informal definition	Number of elements inside a role's boundary in the SR model

Formal definition	`context Role`
	`def:NEIR():Integer = self.hasElement ->`
	`select(e:Element | e.oclIsKindOf(Element)) -> size ()`

Regarding question Q9 (Table 4.15), the value of PSAct (percentage of specific actors) is a measure of the actors' specification. The usage of specialised actor notations such as agents and roles, when the distinction between them is easily made, can help in gaining higher level of detailing in instantiating the stakeholders and capturing the knowledge domain. Lack of use of any of these specialised actor notations might subject the model to lose some useful information. However, excessive use of the special actor notations might lead to much more complex models that might become harder to deal with. Therefore, the choice for the use of the general actor versus the specialised actor notation could be made based on the value and additional information that they will add to the model. Nevertheless, this metric is a useful measure that can be used to compare the specification of actors from different models, and assess whether there is any effect by having a higher precision in the specification level of the actors.

Table 4.15: **Q9** – How specific are the actors?

Symptom	This system uses a significantly different percentage of specific actors, when compared to other systems.
Potential problem	This may be a symptom of an insufficiently detailed system specification, or, conversely, an over-specified one which may be difficult to understand.
Suggested action	Consider scrutinising the types of actors used in the model and reconsidering whether an actor should, or should not be defined using a specific type.
Metric	**PSAct** – *Percentage of Specific Actors*
Informal definition	Percentage of actors with a specific type (agent or role)

Formal definition	`context ISTAR::PSAct()`
	`pre:self.NAct() > 0`
	
	`context ISTAR`
	`def:PSAct():Double = (self.NAgents() +`
	`self.NRoles()) / self.NAct ()`

Requires	**NAgents** – **N**umber of **Agents** (AM A.21) **NRoles** – **N**umber of **Roles** (AM A.22) **NAct** – **N**umber of **Act**ors

Questions Q10 and Q11 (Tables 4.16 and 4.17, respectively) provide different perspectives on the completeness associated with a particular actor. The value of PGWROQ (percentage of an actor's goals with refinement or qualification links), presented in Q10, measures the completeness of goals decompositions associated with an actor. The value of PQWCAQ (percentage of an actor's qualities with contribution or qualification links), presented in Q11, measure the completeness of qualities decompositions associated with an actor. The higher the value of these metrics, the higher the actor's level of completeness. A low number of decomposition of goals and qualities might subject the model to lose some useful information, since the level of precision and detail is lower, leading to understandability problems on how goals and qualities can be achieved.

With respect to question Q12 (Table 4.18), the value of PAWEI (percentage of actors with elements inside its boundary) provide information about how detailed the SR model is with respect to its actors. If there are actors without elements inside, they may not offer any relevant information. If those actors are useful to the model, it is advisable that they are detailed, ergo, having a higher level of completeness.

Concerning questions Q13 and Q14 (Tables 4.19 and 4.20, respectively), the value of PAWOUEI (percentage of actors without unconnected elements inside its boundary) and PAWDOA (percentage of actors with dependency or association links) are measures of how complete the model is and how close we are to finish the modelling process. The higher the value of these metrics, the higher the level of completeness of the model as a whole.

For the goal of evaluating the complexity of iStar 2.0 models, we defined 8 (eight) questions and 23 (twenty-three) metrics, in an average of 3 (three) metrics per question. This goal presents the largest number of questions and metrics. For the goal of evaluating the completeness of iStar 2.0 models, we defined 6 (six) questions and 6 (six) metrics, in

Table 4.16: **Q10** – How detailed are the goals?

Symptom	An actor's goal has an unusually low number of decompositions.
Potential problem	The goal may be under-decomposed.
Suggested action	This goal may be a good candidate for further scrutiny. Consider detailing this goal, if it is under-decomposed. If the actor has no outlier goals, their detail is probably well balanced.
Metric	**PGWROQ** – *Percentage of **G**oals **W**ith **R**efinements **O**r **Q**ualifications*
Informal definition	Percentage of goals with refinement (AND, OR) or qualification links in the SR model

Formal definition	`context ISTAR::PGWROQ()`
	`pre:self.NGIAB () > 0`
	
	`context ISTAR`
	`def:PGWROQ(): Double = (self.NGWD() + self.NGWQ()) / self.NGIAB()`

Requires	**NGWD** – **N**umber of **G**oals **W**ith **D**ecompositions (AM A.23)
NGWQ	**N**umber of **G**oals **W**ith **Q**ualifications (AM A.24)
	NGIAB – **N**umber of **G**oals **I**nside **A**ctors' **B**oundaries (AM A.27)

Table 4.17: **Q11** – How detailed are the qualities?

Symptom	An actor's quality has an unusually low number of decompositions.
Potential problem	The quality may be under-decomposed.
Suggested action	This quality may be a good candidate for further scrutiny. Consider detailing this quality, if it is under-decomposed. If the actor has no outlier quality, their detail is probably well balanced.
Metric	**PQWCOQ** – *Percentage of **Q**ualities **W**ith **C**ontributions **O**r **Q**ualifications*
Informal definition	Percentage of qualities with contribution or qualification links

Formal definition	`context ISTAR::PQWCAQ()`
	`pre:self.NQIAB() > 0`
	
	`context ISTAR`
	`def:PQWCAQ():Double = (self.NQWD() + self.NQWQ()) / self.NQIAB()`

Requires	**NQWD** – **N**umber of **Q**ualities **W**ith **D**ecompositions (AM A.29)
	NQWQ – **N**umber of **Q**ualities **W**ith **Q**ualifications (AM A.30)
	NQIAB – **N**umber of **S**oftgoals **I**nside **A**ctors' **B**oundaries (AM A.33)

an average of 1 (one) metric per question. Overall, we have identified 2 (two) measurement goals, 14 (fourteen) questions that characterise how the goals are achieved, and 29 (twenty-nine) metrics that provide the quantitative information needed to answer the defined questions. A total of 50 (fifty) auxiliary metrics were also defined, which can be found in A.

4.2.3 *i** and iStar 2.0 Tools

In a previous work, we have used Domain Specific Languages mechanisms to develop a tool named iStarLab, which supports the collection and evaluation of metrics about the complexity and completeness of *i** 1.0 models [81, 82]. Although iStarLab is available for use [109], this solution requires the installation of specific software on one's computer. Requirements modelling tools have become less dependent on specific software platforms.

Table 4.18: **Q12** – How detailed is the SR model with respect to its actors?

Symptom	The SR model has an unusually low percentage of actors with elements inside its boundary.
Potential problem	The actors specification may be over-simplistic.
Suggested action	Those actors may be good candidates for further scrutiny. Consider adding and/or detailing elements inside their boundaries. If the system has no outliers, the SR model is probably defined with the typical amount of details, with respect to the elements inside actors boundaries.
Metric	**PAWEI** – *Percentage of Actors With Elements Inside*
Informal definition	Percentage of actors with elements inside its boundary in the SR model

Formal definition	`context ISTAR::PAWEI()`
	`pre:self.NAct() > 0`
	
	`context ISTAR`
	`def:PAWEI():Double = self.NAWEI() / self.NAct()`

Requires	**NAWEI** – **N**umber of **A**ctors **W**ith **E**lements **I**nside (AM A.35) **NAct** – **N**umber of **A**ctors

Table 4.19: **Q13** – How close are we to end the assignment of responsibilities to an actor?

Symptom	The percentage of actors with unconnected elements inside their boundaries represents the percentage of actors with an incomplete specification.
Potential problem	The system specification will not be complete, which can lead to problems in its understandability. This may hamper the developers ability to faithfully implement the system according to the intention of the requirements engineer, because this intention is not documented with enough detail in the requirements model.
Suggested action	Consider completing the specification.
Metric	**PAWOUEI** – *Percentage of Actors WithOut Unconnected Elements Inside*
Informal definition	Percentage of actors without unconnected elements inside its boundary
Formal definition	`context ISTAR` `def:PAWOUEI():Double = 1 - self.PAWUEI()`
Requires	**PAWUEI** – **P**ercentage of **A**ctors **W**ith **U**nconnected **E**lements **I**nside (AM A.37)

Web-based tools do not require any additional installation and can facilitate collaborative access to resources [53]. Taking these advantages into consideration, we developed **2 (two) online tools**, one for *i** 1.0 (Figure 4.2a) and another for iStar 2.0 (Figure 4.2b), both allowing the creation of models, and the automated collection of metrics about those models. Since manually collecting the metrics is time-consuming and error-prone, having a tool that collects this information is essential.

The tools were developed having piStar [169] as a basis, and using web development programming languages and frameworks, namely HTML, CSS, Bootstrap, Javascript, jQuery, JointJS and X-editable. As seen in Subsection 4.2.2, the metrics were initially defined in OCL, upon the meta-models of *i** 1.0 and iStar 2.0. The meta-models and the OCL rules were then transformed and described in Javascript.

Table 4.20: **Q14** – How close are we to end the assignment of links to the actors?

Symptom	The percentage of actors which are not connected to other elements in the system.
Potential problem	The system specification will not be complete, which can lead to problems in its understandability. In particular, the role of the actor in the system may become unclear. This may hamper the developers ability to faithfully implement the system according to the intention of the requirements engineer, because this intention is not documented with enough detail in the requirements model.
Suggested action	Consider completing the specification by creating the necessary associations between the actor and other model elements.
Metric	**PAWDOA** – **P**ercentage of **A**ctors **W**ith **D**ependencies **O**r **A**ssociations
Informal definition	Percentage of actors with dependency or association links

Formal definition	`context ISTAR::PAWDOA()`
	`pre:NAct() > 0`
	
	`context ISTAR`
	`def:PAWDOA():Double = self.NAWDOA() / self.NAct()`

Requires	**NAWDOA** – **N**umber of **A**ctors **W**ith **D**ependencies **O**r **A**ssociations (AM A.47) **NAct** – **N**umber of **A**ctors

4.3 Speed Metrics

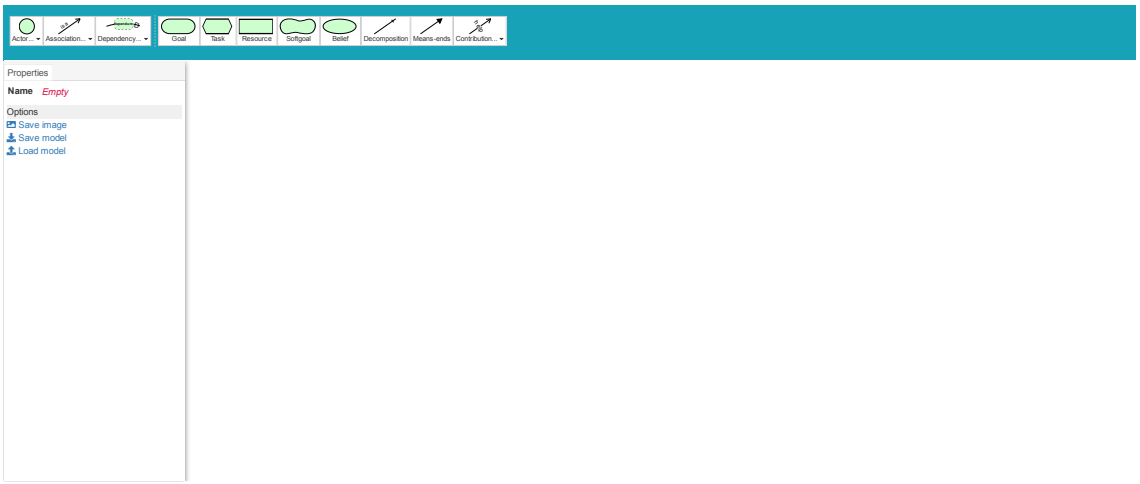
In Table 4.21 we summarise the result of applying the GQM approach to identify a set of metrics that allows satisfying the goal of **speed evaluation**. The first column (*Question*) presents questions that allow evaluating whether the overall goal is being achieved. The second column (*Metric*) shows metrics that provide quantitative information to answer the corresponding question.

Question Q1 is concerned with the overall duration of a task, this is, the time spent by a given stakeholder in a given task. However, a stakeholder may begin to analyse the task at hand before starting providing valid feedback. It may be the case that the stakeholder is analysing and thinking about the task, but not actually performing it. As such, questions Q2 and Q3 are targeted to provide a detailed picture of the moments when the stakeholder really starts and ends providing valid feedback. Finally, Q4 is related with the analysis or reviewing time a stakeholder may take after finishing providing valid feedback A stakeholder may have stopped actively working on the task, but decided to revise it before finishing.

For each metric, we provide a Table containing *(i)* an informal definition, in natural language; and *(ii)* a formal definition using a mathematical expression.

Regarding question Q1 (Table 4.22), the value of *duration* is measured by subtracting the start time to the completion time. The time unit is the *second*, but *minutes* can be used when reporting the results. *Duration* can have values between 0 and, theoretically, infinite seconds, unless a time limit is given to the stakeholder for performing the task.

Concerning question Q2 (Table 4.23), the 2 (two) metrics presented are dependent



(a) *i** 1.0 tool.

(b) iStar 2.0 tool.

Figure 4.2: Tools for the creation of *i** 1.0 and iStar 2.0 models, and the automated collection of metrics about those models. The tools are available online at [102].

Table 4.21: Goal-Question-Metric for the evaluation of speed.

Goal: Evaluate the speed of stakeholders when performing tasks on requirements models	
Question	**Metric**
Q1 – How much time does a stakeholder take to complete a task?	**M1** – Duration
Q2 – How much time does a stakeholder take to start providing valid feedback on a task?	**M2** – First action **M3** – First detection
Q3 – How much time does a stakeholder take to end providing valid feedback on a task?	**M4** – Last action **M5** – Last detection
Q4 – How much time does a stakeholder take between finishing providing valid feedback and considering a task as complete?	**M6** – Processing duration

Table 4.22: **Q1** – How much time does a stakeholder take to complete a task?

Metric	Duration
Informal definition	Time taken by the stakeholder to complete the task, from start to finish.
Formal definition	*completion time – start time*

on the task being performed by the stakeholder, although they are similar in terms of measurement and value. For tasks where an action is required, like the creation or modification of requirements models, *first action* is used. In this case, an action could be, for example, adding the *first* model element to a requirements model. For tasks where a detection is required, like the understanding or reviewing of requirements models, *first detection* is used. In this case, a detection could be, for example, reporting the *first* response element to answer a question about a requirements model. The time unit is the *second*, but *minutes* can be used when reporting the results. If a stakeholder does not perform any action or detection, these metrics are treated as a missing value and removed from all further analysis procedure.

Table 4.23: **Q2** – How much time does a stakeholder take to start providing valid feedback on a task?

Metric	First action
Informal definition	Time taken by the stakeholder to perform the first action on a requirements model.
Formal definition	*first action time – start time*
Metric	First detection
Informal definition	Time taken by the stakeholder to perform the first detection of a model element on a requirements model.
Formal definition	*first detection time – start time*

With respect to question Q3 (Table 4.24) the 2 (two) metrics presented are dependent on the task being performed by the stakeholder, although they are similar in terms of measurement and value. These metrics are dual for the ones presented in Q2. For tasks where an action is required, like the creation or modification of requirements models, *last action* is used. In this case, an action could be, for example, adding the *last* model element to a requirements model. For tasks where a detection is required, like the understanding or reviewing of requirements models, *last detection* is used. In this case, a detection could be, for example, reporting the *last* response element to answer a question about a requirements model. If a stakeholder does not perform any action or detection, these metrics are treated as a missing value and removed from all further analysis procedure.

Finally, with regard to question Q4 (Table 4.25), the value of *processing duration* is measured by subtracting the last action or detection to the total duration of the task. In other terms, it is the time taken by the stakeholder to revise the performed task. The time unit is the *second*, but *minutes* can be used when reporting the results.

Table 4.24: **Q3** – How much time does a stakeholder take to end providing valid feedback on a task?

Metric	Last action
Informal definition	Time taken by the stakeholder to perform the last action on a requirements model.
Formal definition	$last\ action\ time - start\ time$
Metric	Last detection
Informal definition	Time taken by the stakeholder to perform the last detection of a model element on a requirements model.
Formal definition	$last\ detection\ time - start\ time$

Table 4.25: **Q4** – How much time does a stakeholder take between finishing providing valid feedback and considering a task as complete?

Metric	Processing duration
Informal definition	Time taken by the stakeholder to analyse the task after finishing actively working on it.
Formal definition	$duration - last\ (action \vee detection)$

4.4 Visual Ease (Bio)Metrics

In order to assess visual ease, an eye-tracker device (presented in Subsection 2.4.3) is needed to collect the eye-movement of a stakeholder while performing a task. The data can be studied with respect to certain areas of the stimuli, the areas of interest (AOI). Several studies have linked eye-related features to cognitive, mental and memory load [22, 90] as well as emotions [35, 146]. More specifically, a higher number and duration of fixations has been associated with a higher visual attention [171, 194], which is correlated with cognitive processes [55, 75]. The pupil diameter may also vary when an individual experiences high mental and cognitive load [106, 124] or is excited [146].

We used Sharafi et al.'s [193] systematic literature review on the usage of eye-tracking in Software Engineering as the basis for the definition of the metrics. In Table 4.26 we summarise the result of applying the GQM approach to identify a set of metrics that allows satisfying the goal of **visual ease evaluation**. The first column (*Question*) presents questions that allow evaluating whether the overall goal is being achieved. The second column (*Metric*) shows metrics that provide quantitative information to answer the corresponding question.

Questions Q1 and Q2 are concerned with the visual effort of a stakeholder while working on specific AOI, such as relevant (in question Q1) or irrelevant (question Q2) model elements. Questions Q3 and Q4 are targeted at the duration of that visual effort. Finally, questions Q5 and Q6 are related with the search effort, while a stakeholder is trying to find an model element, or exploring the requirements model.

The Eye Tribe device used in this dissertation detects and tracks gaze coordinates. We developed a custom software to collect and store the data from this eye-tracker, including a time stamp and the x and y pixel coordinates of the gaze, if a fixation was detected, the

Table 4.26: Goal-Question-Metric for the evaluation of visual ease.

Goal: Evaluate the visual ease of stakeholders when performing tasks on requirements models	
Question	**Metric**
Q1 - How can we measure the visual effort needed by a stakeholder to process the relevant AOI of a task?	**M1** – Fixation rate on relevant elements
Q2 – How can we measure the visual effort needed by a stakeholder to process the irrelevant AOI of a task?	**M2** – Fixation rate on irrelevant elements
Q3 – How can we measure the average visual attention of a stakeholder on the relevant AOI of a task?	**M3** – Average duration of relevant fixations
Q4 – How can we measure the average visual attention of a stakeholder on the irrelevant AOI of a task?	**M4** – Average duration of irrelevant fixations
Q5 – How can we measure the search effort of a stakeholder while performing a task?	**M5** – Total number of saccades
Q6 – How can we measure the search effort of a stakeholder in the language key of a requirements mode, while performing a task?	**M6** – Total number of saccades to the key

duration of that fixation, and pupils dilatation. We further create the real-time scan-path with the eye trajectory and the detected fixations.

For each metric, we provide a Table containing *(i)* an informal definition, in natural language; and *(ii)* a formal definition using a mathematical expression.

Regarding question Q1 (Table 4.27), the value of the *fixation rate on relevant element* is measured by the number of fixations on relevant AOI among the total number of fixations on the Area of Glance (AOG). Interpreting fixation rate is dependent on the task being performed by the stakeholder. For tasks where an action is required, like the creation or modification of requirements models, a higher fixation rate for a specific AOI indicates that the stakeholders are focused in that AOI. However, it could also indicate that the this area is difficult to modify. For tasks where a detection is required, like the understanding or reviewing of requirements models, smaller rates indicate lower efficiency because the stakeholders spend more time and effort to find the relevant elements.

Table 4.27: **Q1** - How can we measure the visual effort needed by a stakeholder to process the relevant AOI of a task?

Metric	Fixation rate on relevant elements
Informal definition	Rate of a stakeholder's fixations on AOI relevant to the task being performed.
Formal definition	$\frac{number\ of\ fixations\ on\ the\ relevant\ AOI}{number\ of\ fixations\ on\ the\ AOG}$

Concerning question Q2 (Table 4.28), the value of the *fixation rate on irrelevant element*

is measured by the number of fixations on irrelevant AOI among the total number of fixations on the AOG. Interpreting fixation rate is dependent on the task being performed by the stakeholder. For tasks where an action is required, like the creation or modification of requirements models, a higher fixation rate for a specific AOI indicates that the stakeholders are focused in that AOI, not being able to understand its irrelevance. For tasks where a detection is required, like the understanding or reviewing of requirements models, smaller rates indicate higher efficiency because the stakeholders spend less time and effort on irrelevant elements.

Table 4.28: **Q2** – How can we measure the visual effort needed by a stakeholder to process the irrelevant AOI of a task?

Metric	Fixation rate on irrelevant elements
Informal definition	Rate of a stakeholder's fixations on AOI irrelevant to the task being performed.
Formal definition	$\frac{number\ of\ fixations\ on\ the\ irrelevant\ AOI}{number\ of\ fixations\ on\ the\ AOG}$

With respect to question Q3 (Table 4.29), the value of the *average duration of relevant fixations* is measured by the total duration of fixations on relevant AOI among the number of fixations on relevant AOI. This informs us about the time a stakeholder was focused on relevant AOI.

Table 4.29: **Q3** – How can we measure the average visual attention of a stakeholder on the relevant AOI of a task?

Metric	Average duration of relevant fixations
Informal definition	Time of a stakeholder's fixation on an AOI relevant to the task being performed.
Formal definition	$\frac{\Sigma\ duration\ of\ fixations\ on\ the\ relevant\ AOI}{number\ of\ fixations\ on\ the\ relevant\ AOI}$

With regard to question Q4 (Table 4.30), the value of the *average duration of irrelevant fixations* is measured by the total duration of fixations on irrelevant AOI among the number of fixation on irrelevant AOI. This informs us about the time a stakeholder was focused on irrelevant AOI.

Table 4.30: **Q4** – How can we measure the average visual attention of a stakeholder on the irrelevant AOI of a task?

Metric	Average duration of irrelevant fixations
Informal definition	Time of a stakeholder's fixation on an AOI irrelevant to the task being performed.
Formal definition	$\frac{\Sigma\ duration\ of\ fixations\ on\ the\ irrelevant\ AOI}{number\ of\ fixations\ on\ the\ irrelevant\ AOI}$

Concerning question Q5 (Table 4.31), the value of *total number of saccades* is measured by summing all the saccades in a stimulus. A higher number of saccades indicates more searching effort.

Table 4.31: **Q5** – How can we measure the search effort of a stakeholder while performing a task?

Metric	Total number of saccades
Informal definition	Number of saccades a stakeholder does while performing a task.
Formal definition	Σ *saccades*

Lastly, regarding question Q6 (Table 4.32), the value of *total number of saccades to the key* is measured by summing all the saccades a stakeholder performs to a specific AOI in a stimulus. In this case, the AOI is the language key of the requirements models, available when a stakeholder is performing a task on that model. A higher number of saccades to the key indicates doubt and need to verify the meaning of a model element.

Table 4.32: **Q6** – How can we measure the search effort of a stakeholder in the language key of a requirements mode, while performing a task?

Metric	Total number of saccades to the key
Informal definition	Number of saccades a stakeholder does to the language key of a requirements model while performing a task.
Formal definition	Σ *saccades to the key AOI*

With the same eye-tracking information (fixations, saccades, and scan-paths), there are a variety of metrics and visualisations that can be computed. Examples include spacial density, convex hull, attention switching frequency, gaze-plots, among other. However, they are **not** covered in the context of this dissertation, as they were not essential for answering our research questions.

4.5 Mental Ease (Bio)Metrics

In order to assess mental ease, an EEG scanner (presented in Subsection 2.4.4) is needed to record the electrical activity of the brain and the brain waves, which can be divided into frequency bands. All the metrics are computed by an evaluation of those frequency bands. Several studies have linked those frequency bands with various cognitive and emotional states, including mental workload [22, 184], arousal [180], and happiness or sadness [133, 149, 150]. The ratios of the frequency bands have also been linked with memory load [87], task engagement [15, 127], and arousal [133]. More specifically, a decrease of alpha and often an increase in theta waves indicates an increase in attention demand and working memory load [71, 150, 210]. Alpha waves typically occur when an individual is in a relaxed state. As soon as the mental or physical activity level increases, the alpha waves disappear or their amplitude gets significantly smaller. Gamma waves often appear as a reaction to a sensory stimuli, and beta waves occur when performing mental or physical activities. Finally, theta waves appear when an individual experiences (dis)pleasure [3].

In Table 4.33 we summarise the result of applying the GQM approach to identify a set of biometrics that allows satisfying the goal of **mental ease evaluation**. The first column (*Question*) presents questions that allow evaluating whether the overall goal is being achieved. The second column (*Metric*) shows metrics that provide quantitative information to answer the corresponding question.

Question Q1 is concerned with the focus or attention of a stakeholder on a task while performing it. Question Q2 is targeted at the mental activity. Finally, question Q3 is related with the learning process and memory accessing.

Table 4.33: Goal-Question-Metric for the evaluation of mental ease.

Goal: Evaluate the mental ease of stakeholders when performing tasks on requirements models	
Question	**Metric**
Q1 – How can we measure the focus of a stakeholder while performing a task?	**M1** – Average attention
Q2 – How can we measure the mental effort of a stakeholder while performing a task?	**M2** – Average mental workload
Q3 – How can we measure the relative level of understanding of a stakeholder while performing a task?	**M3** – Average familiarity

The NeuroSky MindWave EEG Headset used in this dissertation collects brain waves and signals mainly from the pre-frontal cortex. We compute the power spectrum distribution for each of the brain wave frequency bands. Since every person has a unique power spectrum distribution, we compute the ratio of each band with one another in order to compare the values between individuals. In addition, we compute the *average attention*, *mental workload* and *familiarity* of a stakeholder while (s)he was performing the task.

For each metric, we provide a Table containing *(i)* an informal definition, in natural language; and *(ii)* a formal definition using a mathematical expression.

Regarding question Q1 (Table 4.34), the value of *average attention* is measured by taking into account the attention values per millisecond a stakeholder has in the course of a task. The computed *average attention* can have values between 0 an 1. The attention level increases when the stakeholder focuses on a single a task element, and decreases when distracted.

Table 4.34: **Q1** – How can we measure the focus of a stakeholder while performing a task?

Metric	Average attention
Formal definition	Intensity of mental focus or attention of the stakeholder while performing performing a task.
Formal definition	$\frac{\Sigma\ attention\ value\ per\ ms}{total\ duration\ in\ ms}$

Concerning question Q2 (Table 4.35), the value of *average mental workload* is measured

by taking into account the mental workload values per millisecond a stakeholder has in the course of a task. The computed *average mental workload* can have values between 0 and 1. The harder a stakeholder's brain works on a task, the higher the value.

Table 4.35: **Q2** – How can we measure the mental effort of a stakeholder while performing a task?

Metric	Average mental workload
Informal definition	Intensity of mental effort of the stakeholder while performing a task.
Formal definition	$\frac{\Sigma\ mental\ workload\ value\ per\ ms}{total\ duration\ in\ ms}$

With respect to question Q3 (Table 4.36), the value of *average familiarity* is measured by taking into account the familiarity values per millisecond a stakeholder has in the course of a task. The computed *average familiarity* can have values between 0 and 1. The higher a stakeholder is remembering previous or recently obtained knowledge, the higher the value.

Table 4.36: **Q3** – How can we measure the relative level of understanding of a stakeholder while performing a task?

Metric	Average familiarity
Informal definition	Intensity of understand, learning, and memory access of a stakeholder while performing a task.
Formal definition	$\frac{\Sigma\ familiarity\ value\ per\ ms}{total\ duration\ in\ ms}$

We can also use the frequency bands to calculate meditation (level of mental calmness or relaxation), appreciation (level of enjoyment), creativity (level of innovative and creative thinking), among others. However, these metrics are **not** covered in the context of this dissertation.

4.6 Emotional Ease (Bio)Metrics

In order to assess emotional ease, an EDA scanner (presented in Subsection 2.4.5) is needed to record the electrical conductance of the skin, as well as the heart rate. Several studies have linked a higher average skin conductive level with a greater cognitive load, task difficulty, and stress [157, 201]. Electrodermal activity correlates strongly with arousal [21, 190], stress and anxiety [89, 141]. For the heart rate variability, features representing the difference in time between two heart beats, such as RMSSD (root mean square of successive differences) or NN50 (the number of pairs of successive beat-to-beat intervals that differ more than 50ms) have been linked to task difficulty and stress [4, 235]. An increase in the heart rate, when in a stationary state, can be related with anxiety [54, 137] and stress [208, 244].

In Table 4.37 we summarise the result of applying the GQM approach to identify a set of metrics that allows satisfying the goal of **emotional ease evaluation**. The first

column (*Question*) presents questions that allow evaluating whether the overall goal is being achieved. The second column (*Metric*) shows metrics that provide quantitative information to answer the corresponding question.

Question Q1 is concerned with the emotions a stakeholder may feel while performing a task on a requirements model.

Table 4.37: Goal-Question-Metric for the evaluation of emotional ease.

Goal: Evaluate the emotional ease of stakeholders when performing tasks on requirements models	
Question	**Metric**
Q1 – How can we measure the emotional activation (excitement or stress) of a stakeholder while performing a task?	**M1** – Average skin conductive level **M2** – Average RMSSD **M3** – Average NN50

The BioSignalsPlux Wristband used in this dissertation collects the skin's electrical activity and heart rate, automatically producing a report with the average skin conductive level, and hear rate variability for both RMSSD and NN50.

4.7 Perceived Effort Metrics

For evaluating the *perceived effort* a stakeholder experiences while performing a task, we are using the NASA-TLX (presented in Subsection 2.4.6). The questions, and corresponding metrics, are the ones available in the official online NASA-TLX documentation [227].

In Table 4.38 we summarise the result of applying the GQM approach to identify a set of metrics that allows satisfying the goal of **perceived effort evaluation**. The first column (*Question*) presents questions that allow evaluating whether the overall goal is being achieved. The second column (*Metric*) shows metrics that provide quantitative information to answer the corresponding question.

NASA-TLX can be administrated by using the official paper and pencil version [153], as well as through the official Apple iOS application [155]. There are also various unofficial computerized implementations of the NASA-TLX, such as a version by Professor Keith Vertanen (Michigan Technological University) [154]. We have adapted this web-based version in our quasi-experiments.

4.8 Summary

We followed the GQM approach to propose a set of (bio)metrics for the evaluation of requirements models. In total, we have defined 8 goals, 34 questions and 50 metrics. Some of the metrics are available in the literature (with the reference provided), while others were defined as part of this dissertation. We have also developed a web-based tool

Table 4.38: Goal-Question-Metric for the evaluation of perceived effort, adapted from [93, 94].

Goal: Evaluate the perceived effort of stakeholders when performing tasks on requirements models	
Question	**Metric**
Q1 – How much mental and perceptual activity was perceived by the stakeholder while performing the task?	**M1** – (Perceived) mental demand
Q2 – How much physical activity was perceived as required by the stakeholder while performing the task?	**M2** – (Perceived) physical demand
Q3 – How much time pressure was perceived by the stakeholder while performing the task?	**M3** – (Perceived) temporal demand
Q4 – How successful does the stakeholder think (s)he was in accomplishing the goals of the task?	**M4** – (Perceived) performance
Q5 –How hard the stakeholder consider (s)he had to work (mentally and physically) to accomplish his/hers level of performance?	**M5** – (Perceived) effort
Q6 – How insecure, discouraged, irritated, stressed or annoyed the stakeholder felt while performing the task?	**M6** – (Perceived) frustration

that allows the creation of *i** models and automatically collects complexity and completeness metrics about those models. In combination, these (bio)metrics can give meaningful insights about requirements models and the way stakeholders interact with them.

Chapter 5

A Family of 16 Quasi-Experiments for the Evaluation of Requirements Models

In this dissertation, we propose an integrated methodology for the learnability and appropriateness recognisability evaluation applied to *i** (*i** 1.0 and iStar 2.0), and use cases (ARNE and ALCO templates), by performing a family of 16 quasi-experiments with different types of participants. In this Chapter, we present the experimental protocol used in all the conducted quasi-experiments, following Jedlitschka et al.'s guidelines [113] on how to report experiments and quasi-experiments in Software Engineering. The Chapter covers planning, execution, and analysis steps that are common to all the quasi-experiments, thus presenting an overview of the empirical research performed. The discussion on the results and their implications, as well as further details on the particularities of each quasi-experiment, are described in the specific Chapters of the studies (see Chapters 6, 7 and 8). However, we described the threats to validity in this Chapter, as they are common to all the quasi-experiments performed. Finally, we present a web-based replication package, that aims to facilitate independent replications of the quasi-experiments described.

5.1 Experiments Planning

We performed a family of 16 (sixteen) quasi-experiments, presented in Figure 5.1. We executed 8 evaluations for learnability and 8 for appropriateness recognisability, where 4 were about *i** and 4 about use cases, in both situations. We designed separate quasi-experiments, with any given participant only performing one of them. A discussion on this design choice is available in Section 5.1.6. For each quasi-experiment, we first performed a pilot and made adjustments accordingly. The actual studies took place at our University, in the Informatics department (DI-FCT-UNL); and at 11 (eleven) Portuguese software companies. The companies are not named due to an anonymity agreement.

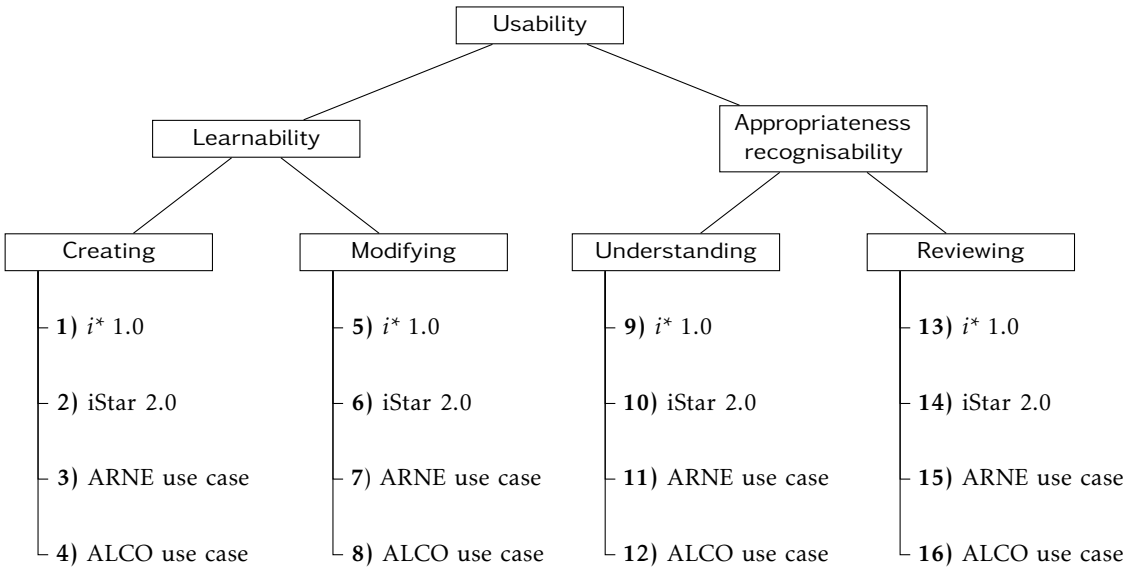


Figure 5.1: Family of 16 quasi-experiments for the usability evaluation of *i** and use cases.

5.1.1 Goals

We describe our research goals using the GQM research goal template [7, 8]:

> **Analyse** differences in *[set]*, **for the purpose of** evaluation, **with respect to** their effects on the *[task]* of *[artefact]*, **from the viewpoint of** researchers, **in the context of** quasi-experiments conducted at our University and at software companies.

Our goals are related with differences in the following ***[sets]***: (*i*) *i** versions, with *i** 1.0 *versus* iStar 2.0; (*ii*) use cases templates, with ARNE *versus* ALCO; and (*iii*) levels of the GenderMag facets, with Abby *versus* Tim.

The ***[task]*** can be of the following types: (*i*) creation; (*ii*) modification; (*iii*) understanding; and (*iv*) review. We further break down these goals into sub-goals, concerning the effects of the different *[set]*, in terms of *accuracy*, *speed*, and *ease*. The refined goals are obtained by replacing *[task]* with *accuracy to [task]*, *speed to [task]*, and *ease to [task]*.

Finally, the ***[artefacts]*** are the following: (*i*) *i** 1.0; (*ii*) iStar 2.0; (*iii*) ARNE use case template; and (*iv*) ALCO use case template.

The concrete goals for each of the studies are described in the specific Chapters 6, 7 and 8. However, for illustrating the replacements, we define one of our goals:

> **(GN1) Analyse** differences in the *i** versions, **for the purpose of** evaluation, **with respect to** their effects on the creation of *i** SR models, **from the viewpoint of** researchers, **in the context of** experiments conducted at our University and at software companies.

5.1.2 Participants

Our main target population are current or future computer scientists in general, and requirements engineers in particular. However, since requirements models are used for communication with different types of stakeholders, we were also interested in a variety of people, covering professional backgrounds in both sciences and social sciences. This gives us a broader perspective on how different people interact with requirements artefacts and, for instance, understand if a particular type of artefact is better suited for people with a given professional background and level of experience.

The participants in all the quasi-experiments were recruited through **convenience** and **snowball sampling**, both non-probability sampling techniques. In the former, participants are selected due to their convenient accessibility and proximity to the researcher; while the latter involves asking people who have already participated to nominate other people they believe would be willing to be part of the study [123].

Random sampling occurs when all members of the target population have an equal chance of being selected to participate. As a result, the conclusions of the study can be generalisable. The main problem is that random sampling is rather difficult and has a high cost. In our particular case, it would imply having a population register, with a list of computer scientists or requirements engineers. To the best of our knowledge, there is no such list. Convenience sampling, on the other hand, is fast, easy, an less expensive [245]. In Software Engineering, convenience sampling is one of the most common ways of recruiting participants [207]. However, the sample may not be representative of the target population. Nevertheless, we argue that our sample would not greatly differ from an ideal sample that was randomly selected. We cover both students and practitioners, that studied at different Universities, and with a diverse background and experience.

At our University, participants were made aware of the study, either by direct communication or by e-mail, and volunteered to participate. Several of them have conducted, or will conduct in a near future, studies in the context of their research projects, so motivating them to participate was not a problem. Some of these participants actively recruited their contacts to participate, hence the snowball sampling. This technique also allowed us to have a more diversified set of participants.

For the studies with practitioners, in a real-world environment, the first contact with the company was possible by leveraging personal contacts. Then, the company employees were made aware of the study and volunteered to participate. Our Department has a close relationship with several Portuguese software companies, mainly due to internships for the third-year students, and Master's theses with industry collaboration. Furthermore, several employees of these companies are alumni of our University. As such, collaboration between the Department and software companies is perceived as important by both counterparts. Larger companies also provided us a multidisciplinary environment, and participants with diverse characteristics, backgrounds, and professional experiences.

Except for the iStar 2.0 creation and modification quasi-experiments, where we had 50

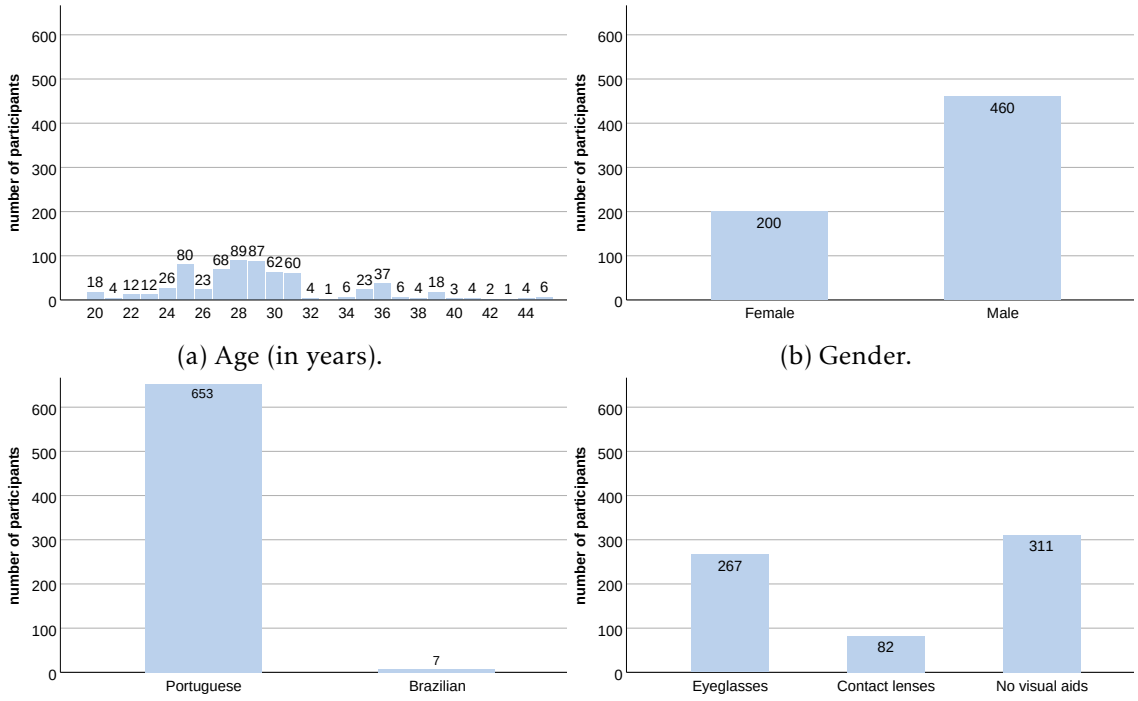
participants each, all the other experiments were performed by 40 participants. In total, we had 660 participants. We calculated the sample size needed to ensure an adequate power level, where 0.8 is considered appropriate (80% probability of correctly detecting a real effect) [122]. We chose a standardised large Cohen's effect size for $\alpha = 0.05$ (significance level). To detect a large difference between two independent sample means at $\alpha = 0.05$, 26 participants are required in each group [43]. A medium difference would require 64 participants in each group, in a total of 1024 for the 16 quasi-experiments. We decided to have as many participants as possible, after the initial 26 per group, knowing it possibly would not be feasible to have 1024, in total, to achieve the amount for medium difference.

No compensation was given to participants for performing the study. We had no mortality of the participants. This means that none of the participants refused to answer any question, nor decided to withdraw from the study. However, some data of a small set of participants was removed from the analysis. Further details, such as the cause of the removal, and potential impact on the results, are discussed in Section 5.2.3.

From all the participants, we collected demographic data on *age*, *gender*, *nationality*, *usage of reading devices* (eyeglasses, or contact lenses), *field of studies*, *highest completed level of education*, *current level of education*, *current occupation*, the previous *experience* with the artefact they used in their task, including *usage time* and *last use*, and their *knowledge on other requirements models*. For comparison purposes, all the following bar charts related with demographic information have the same vertical scale (a maximum of 600 participants). However, this makes some bar charts harder to individually analyse, since the number of participants is significantly lower than the scale used. As such, and due to its high number, we present the charts with their specific scale in a webpage [213].

Concerning participants *age* distribution (Figure 5.2a), they had between 20 and 45 years old, with an average of 28 years old. With respect to *gender* (Figure 5.2b) there were 460 male participants and 200 females. In terms of *nationality* (Figure 5.2c), 653 were Portuguese and 7 were Brazilian. Regarding the *usage of reading devices* (Figure 5.2d), 267 participants wore eyeglasses and 82 had contact lenses.

All participants had some university level training. Their *field of studies* (Figure 5.3a) spanned across multiple areas. We had 1 biomedical engineer (BE), 399 computer scientists (CS), 17 designers (D), 24 electrotechnical engineers (EE), 68 environmental engineers (EnvE), 1 forensic scientist (FS), 51 historians (H), 1 information technologist (IT), 55 lawyers (L), 2 mechanical engineers (ME), and 41 medical doctors (MD). For *highest completed level of education* (Figure 5.3b), 92 completed high school, 254 concluded a BSc, 303 had a MSc, and 11 a PhD degree. Concerning *current level of education* (Figure 5.3c), 8 were in the first year of the BSc degree, 26 on the second year, and 60 on the third and final year. As for MSc students, 88 were in the first year, and 82 were on the second and final year. Finally, 82 were doing a PhD, 3 were doing a Post-Doc, and 282 were no longer studying. The ones that were no longer studying had at least 4 years of experience. With respect to *current occupation* (Figure 5.3d), 244 of the participants were students, 125



(a) Age (in years).

(b) Gender.

(c) Nationality.

(d) Usage of reading devices.

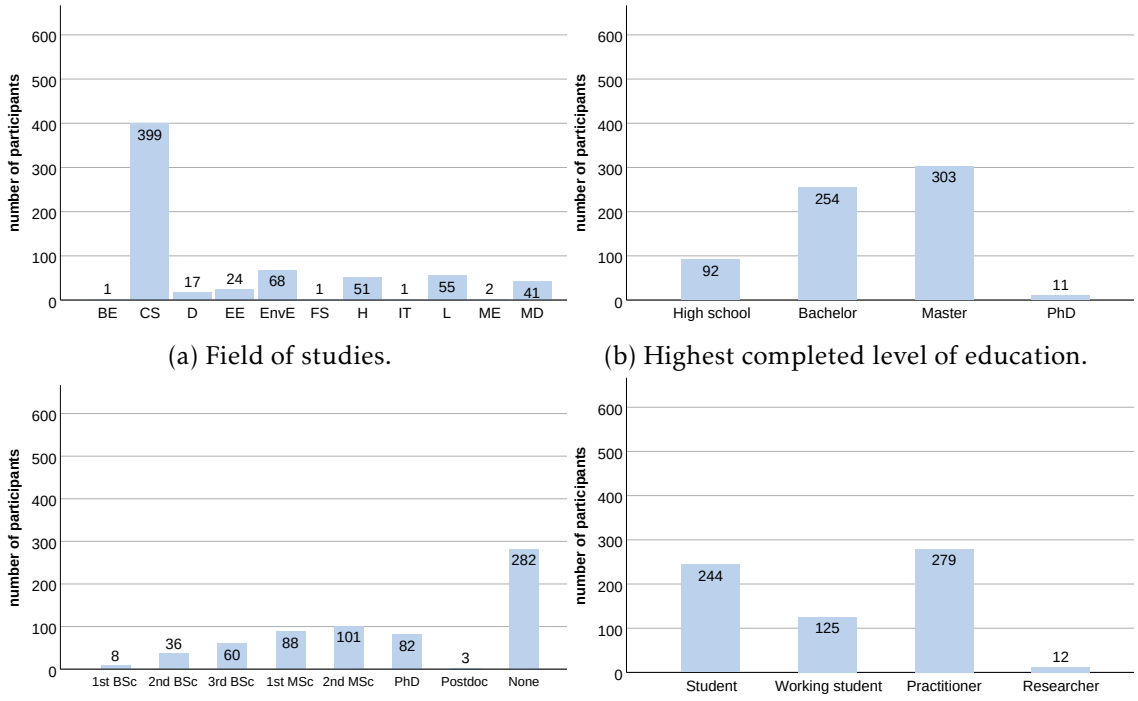
Figure 5.2: Participants general demographic information.

were working students, 279 were practitioners, and 12 were researchers.

Regarding previous *experience* (Figure 5.4a) with the artefact used in the task, for 431 participants it was their first contact with the artefact. However, 168 learnt it in the context of a course, and 61 in a professional environment. In those two latter scenarios, participants *usage time* with the artefacts (Figure 5.4b) had an average of 5 months. Participants tend to refer to the last usage time in terms of full years (for example, saying one or two years ago, and never one and a half years ago). On the other hand, some participants referred to 3, 4 or 6 months. We argue that all those months correspond to an University semester, depending on how people count. As for the *last use* of the artefact (Figure 5.4c), although the majority of participants were no longer using the artefact, 33 participants were still using it in their daily work when the studies were conducted. Lastly, in terms of *knowledge on other requirements models* (Figure 5.4d), 302 participants claim to know UML in general, 50 referred to BPMN, 11 specifically said to work with flowcharts in particular, 4 mentioned KAOS, and 1 BPEL. The remaining 292 participants didn't report knowing any requirements language.

Participants spanned a reasonably wide range of values of each of the GenderMag facets, with 12 participants being characterised as a "pure" Abby and 27 as a "pure" Tim (Figures 5.5a and 5.5b). The other 621 participants had mixed characteristics of both Abby and Tim.

When analysing each facet (Figure 5.5c), the majority of the participants was identified as Tim in the *motivation*, *risk*, and *learning style* facets. For *information processing* and *self*



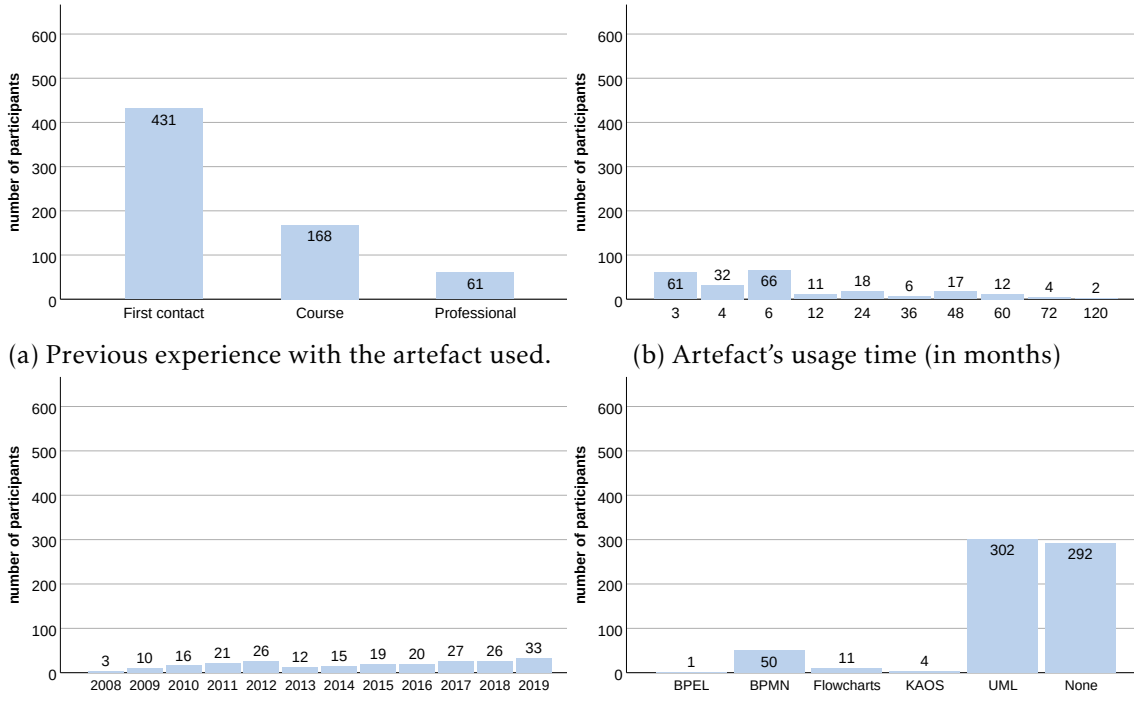
(a) Field of studies.

(b) Highest completed level of education.

(c) Current level of education.

(d) Current occupation.

Figure 5.3: Participants academic and professional demographic information.



(a) Previous experience with the artefact used.

(b) Artefact's usage time (in months)

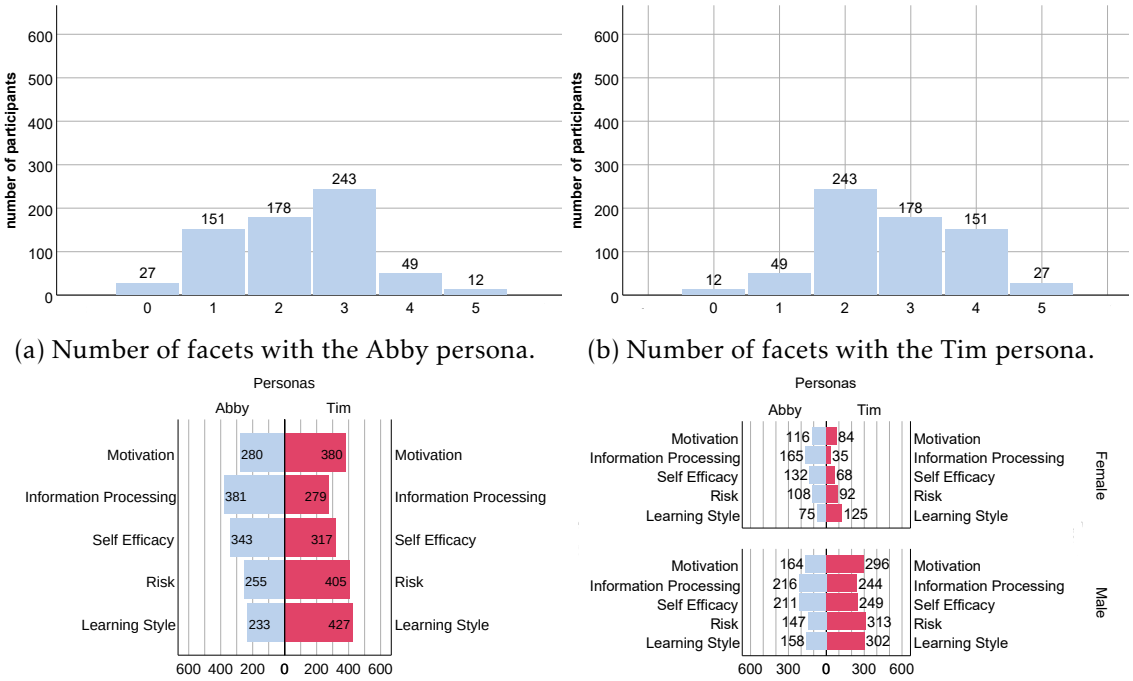
(c) Artefact's last year of use

(d) Knowledge on other requirements models

Figure 5.4: Participants knowledge on requirements models.

efficacy, on the other hand, the majority of participants was described as Abby.

Taking a closer look into the relationship between the persona in each of the facets and the gender of participants (Figure 5.5d), the majority of female participants was characterised as Abby in all the facets, being *learning style* an exception. As for the males, the majority of participants participants was classified as Tim in all the facets. These results support the literature claim [10, 191] that characteristics in how people solve problems often cluster by gender.



(a) Number of facets with the Abby persona.

(b) Number of facets with the Tim persona.

(c) Personas in each facet.

(d) Personas in each facet, by gender.

Figure 5.5: Participants distribution across GenderMag facets.

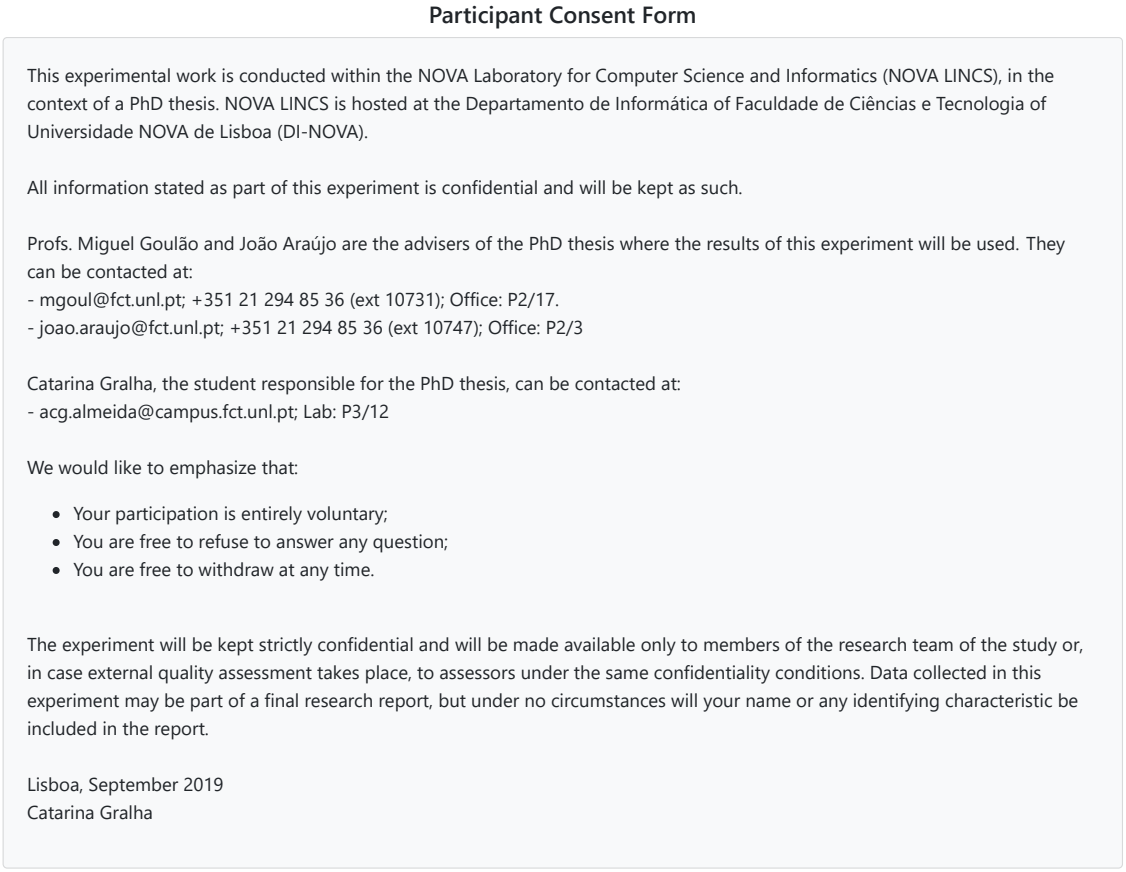
The demographic information for the participants of each quasi-experiment is described in the specific Chapters 6 and 7.

5.1.3 Experimental Materials

The experimental materials for all the quasi-experiments included *(i)* a participant consent form; *(ii)* a video of fish swimming; *(iii)* a video tutorial about the language in which the artefact was specified; *(iv)* a problem description for the creation task; a problem description, an initial model and a new requirement for the modification task; a model and a set of questions for the understanding task; or a model with semantic defects for the review task; *(v)* a NASA-TLX questionnaire; *(vi)* a demographic questionnaire; and *(vii)* a GenderMag questionnaire.

The **participant consent form**, adapted from [183] and presented in Figure 5.6, explained that the participation was entirely voluntary, the participants could refuse to

answer any question and could leave at any time, and that all the collected data would remain anonymous.



Participant Consent Form

This experimental work is conducted within the NOVA Laboratory for Computer Science and Informatics (NOVA LINCS), in the context of a PhD thesis. NOVA LINCS is hosted at the Departamento de Informática of Faculdade de Ciências e Tecnologia of Universidade NOVA de Lisboa (DI-NOVA).

All information stated as part of this experiment is confidential and will be kept as such.

Profs. Miguel Goulão and João Araújo are the advisers of the PhD thesis where the results of this experiment will be used. They can be contacted at:
- mgoul@fct.unl.pt; +351 21 294 85 36 (ext 10731); Office: P2/17.
- joao.araujo@fct.unl.pt; +351 21 294 85 36 (ext 10747); Office: P2/3

Catarina Gralha, the student responsible for the PhD thesis, can be contacted at:
- acg.almeida@campus.fct.unl.pt; Lab: P3/12

We would like to emphasize that:

- Your participation is entirely voluntary;
- You are free to refuse to answer any question;
- You are free to withdraw at any time.

The experiment will be kept strictly confidential and will be made available only to members of the research team of the study or, in case external quality assessment takes place, to assessors under the same confidentiality conditions. Data collected in this experiment may be part of a final research report, but under no circumstances will your name or any identifying characteristic be included in the report.

Lisboa, September 2019
Catarina Gralha

Figure 5.6: Participant consent form.

The **video of fish swimming**, with a snapshot presented in Figure 5.7, is 2 minutes long, and served as a baseline to normalise the captured biometrics data [68, 147]. It also helped participants to relax and better focus on the task at hand.

The **video tutorial** explained the concepts of the modelling language the participant would interact with in the subsequent task, at the same time it illustrated those same concepts by creating an *i** SR model or a use case specification about a meeting scheduler system. Further details on these tutorials are available in the specific Chapters 6 and 7.

The **tasks** were related with the modelling language the participant saw the tutorial on. They could be creation, modification, understanding or reviewing tasks. Further details on these tasks are available in the specific Chapters 6 and 7.

The **NASA-TLX questionnaire**, presented in Figure 5.8, collected feedback on the participants' perceptions with respect to effort on the performed task.

The **demographic questionnaire**, presented in Figure 5.9, collected the demographic information on the participants.

The **GenderMag questionnaire**, presented from Figure 5.10 to 5.13, has a set of 9-point Likert questions. In total, there are 20 questions, divided into groups related

Figure 5.7: Snapshot of the video of fish swimming at a fish tank.

NASA Task Load Index

Workload measures

Click on each scale at the point that best indicates your experience with the task

Mental Demand	Low — High	How much mental and perceptual activity was required (e.g. thinking, deciding, calculating, remembering, looking, searching, etc)? Was the task easy or demanding, simple or complex, exacting or forgiving?
Physical Demand	Low — High	How much physical activity was required (e.g. pushing, pulling, turning, controlling, activating, etc)? Was the task easy or demanding, slow or brisk, slack or strenuous, restful or laborious?
Temporal Demand	Low — High	How much time pressure did you feel due to the rate of pace at which the tasks or task elements occurred? Was the pace slow and leisurely or rapid and frantic?
Performance	Good — Poor	How successful do you think you were in accomplishing the goals of the task set by the experimenter (or yourself)? How satisfied were you with your performance in accomplishing these goals?
Effort	Low — High	How hard did you have to work (mentally and physically) to accomplish your level of performance?
Frustration	Low — High	How insecure, discouraged, irritated, stressed and annoyed versus secure, gratified, content, relaxed and complacent did you feel during the task?

Figure 5.8: NASA-TLX workload measure for participants' effort perceptions.

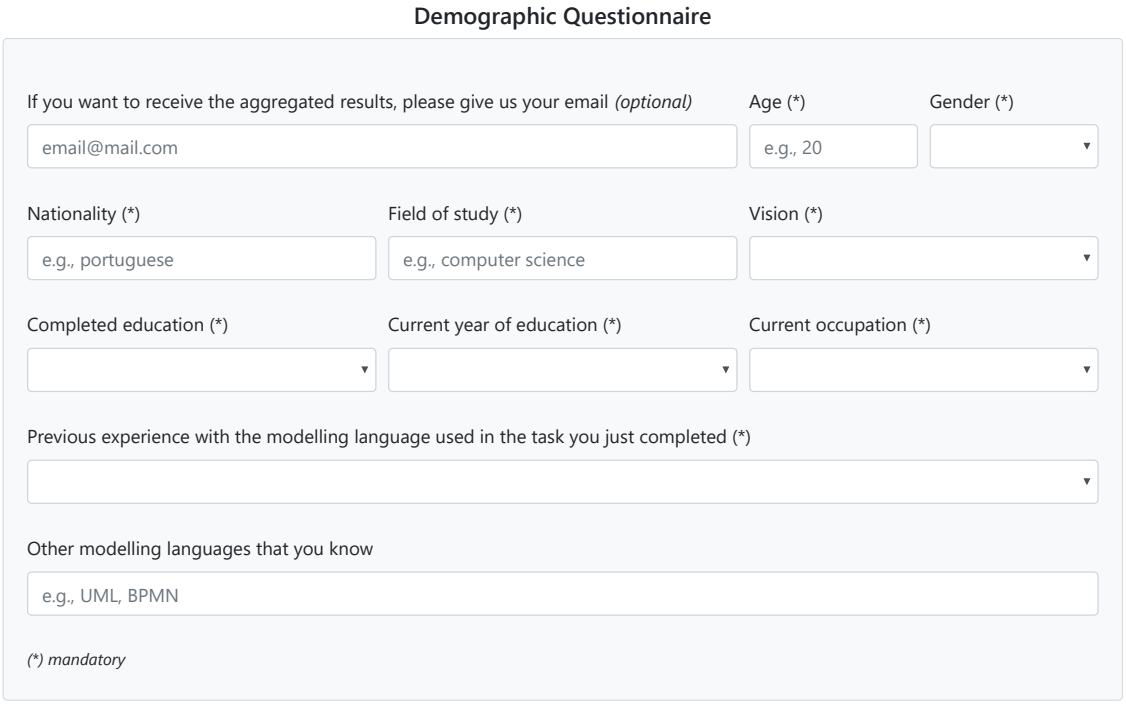


Figure 5.9: Demographic questionnaire.

with each one of the facets. The scores for each facet are added, and each individual is compared to the grand median (median of medians) for that facet. If a participant is above the median on a given facet, we name him/her Tim (on that facet alone). If s(he) is below, we name him/her Abby (on that facet alone). Pats are the ones that are exactly in the grand median. However, due to way scores are calculated, Pats are rarely present in the facets [236].

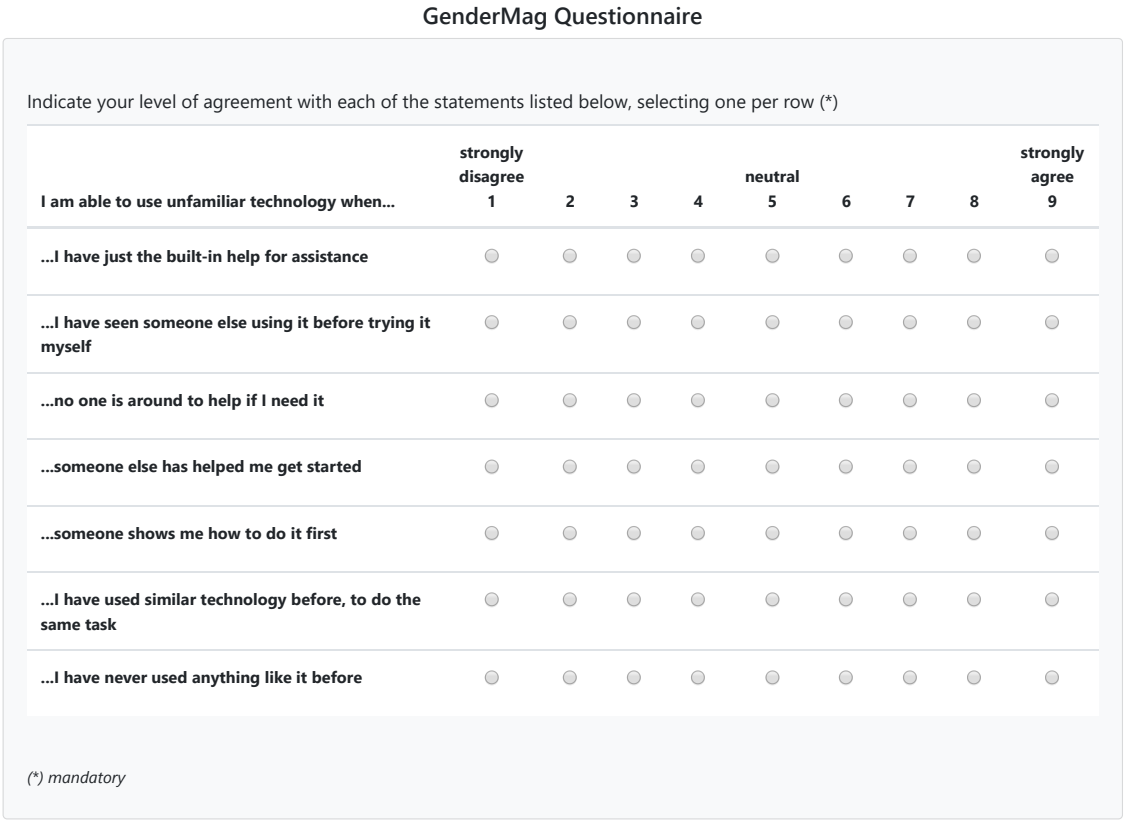
In all the tasks, every element presented to participants was comfortably readable in the 22 **inch monitor** used to conduct the experiment.

5.1.4 Tasks

In all the tasks, the domain was a **booking management system for an hotel**. We opted for a relatively known domain in order to reduce the effect of the results being related with difficulties in understanding the domain itself, and not due to the artefacts that were under study. However, we are aware that tacit knowledge may also play an important role in the performance of the participants.

Each participant completed 1 (one) task. However, there were 4 (four) types of tasks (*creating*, *modifying*, *understanding* and *reviewing*) for each of the 4 (four) 🖹 artefacts (*i* 1.0*, *iStar 2.0*, *ARNE use cases* and *ALCO use cases*), in a total of 16 (sixteen) tasks.

In the **creation task**, participants had to create an *[🖹 artefact]* given a small problem description. In the **modification task**, participants had to modify an initial *[🖹 artefact]*,



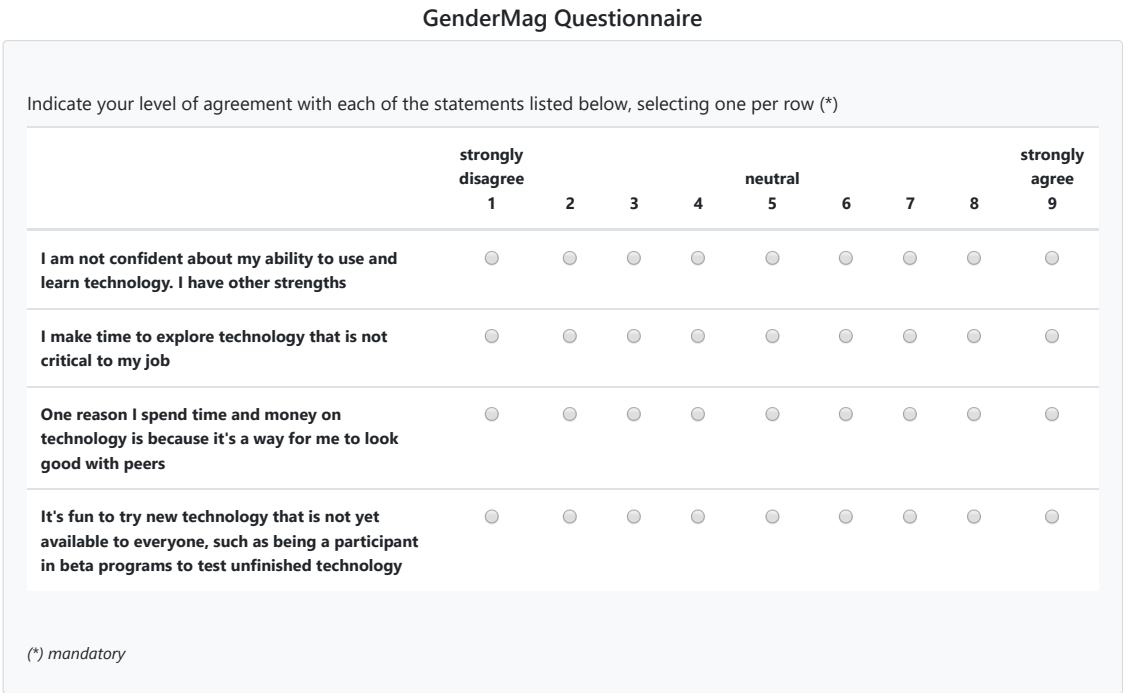
GenderMag Questionnaire

Indicate your level of agreement with each of the statements listed below, selecting one per row (*)

I am able to use unfamiliar technology when...	strongly disagree 1	2	3	4	neutral 5	6	7	8	strongly agree 9
...I have just the built-in help for assistance									
...I have seen someone else using it before trying it myself									
...no one is around to help if I need it									
...someone else has helped me get started									
...someone shows me how to do it first									
...I have used similar technology before, to do the same task									
...I have never used anything like it before									

(*) mandatory

Figure 5.10: GenderMag questionnaire: part 1.



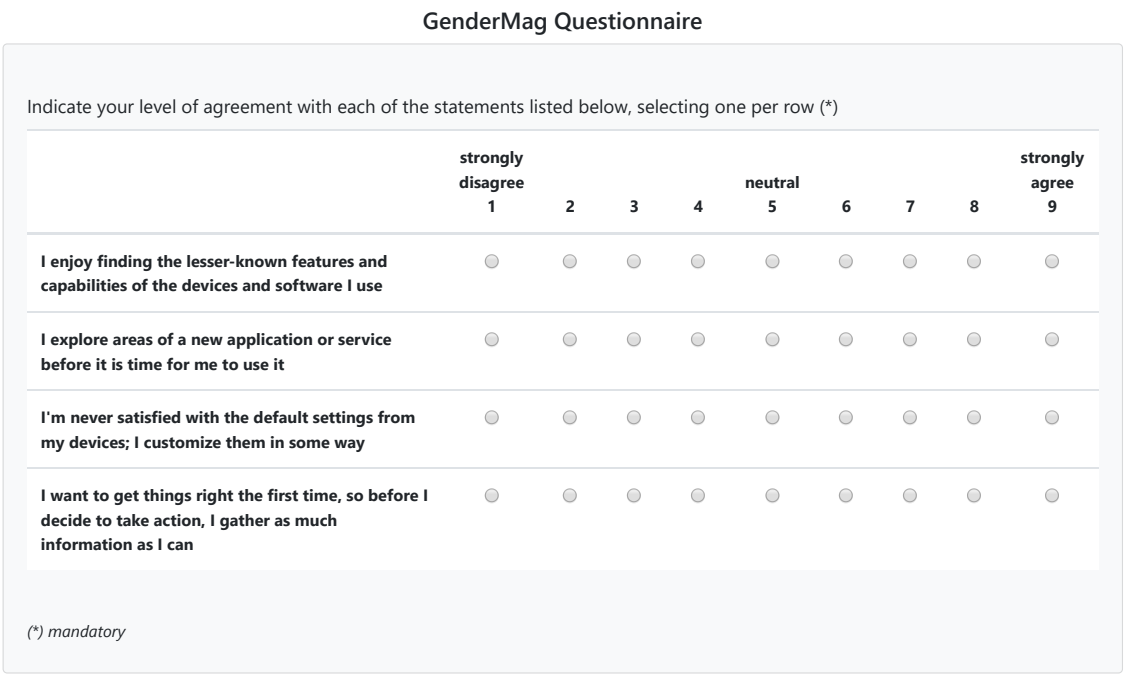
GenderMag Questionnaire

Indicate your level of agreement with each of the statements listed below, selecting one per row (*)

	strongly disagree 1	2	3	4	neutral 5	6	7	8	strongly agree 9
I am not confident about my ability to use and learn technology. I have other strengths									
I make time to explore technology that is not critical to my job									
One reason I spend time and money on technology is because it's a way for me to look good with peers									
It's fun to try new technology that is not yet available to everyone, such as being a participant in beta programs to test unfinished technology									

(*) mandatory

Figure 5.11: GenderMag questionnaire: part 2.



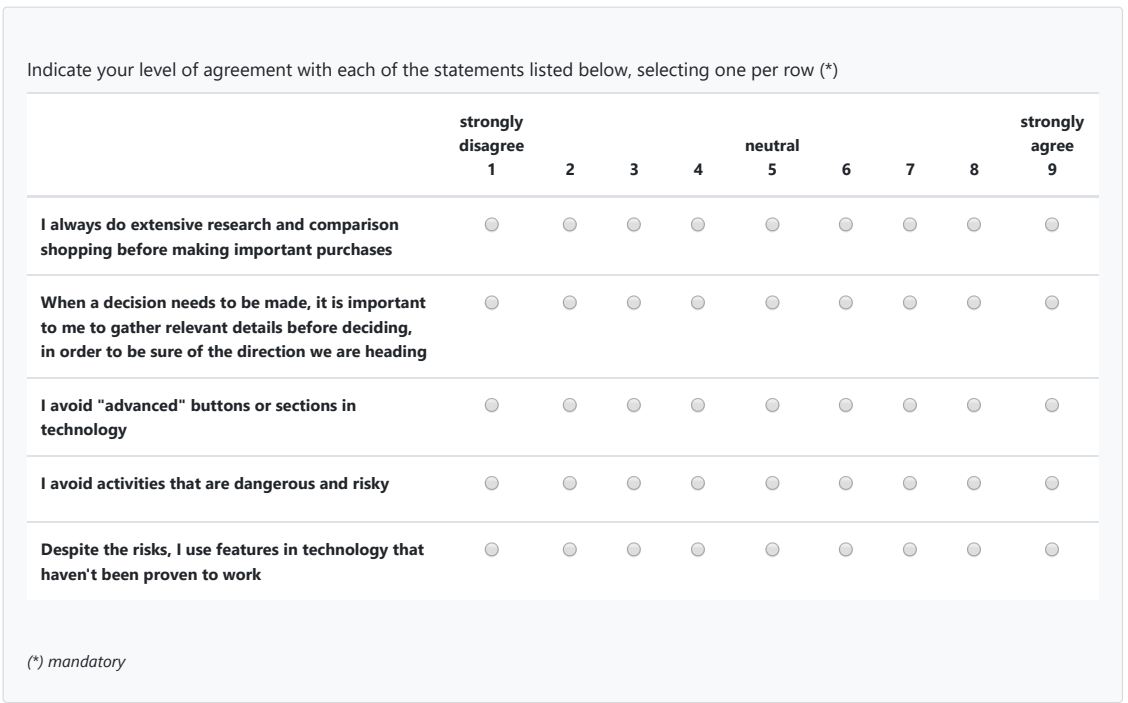
GenderMag Questionnaire

Indicate your level of agreement with each of the statements listed below, selecting one per row (*)

	strongly disagree 1	2	3	4	neutral 5	6	7	8	strongly agree 9
I enjoy finding the lesser-known features and capabilities of the devices and software I use	○	○	○	○	○	○	○	○	○
I explore areas of a new application or service before it is time for me to use it	○	○	○	○	○	○	○	○	○
I'm never satisfied with the default settings from my devices; I customize them in some way	○	○	○	○	○	○	○	○	○
I want to get things right the first time, so before I decide to take action, I gather as much information as I can	○	○	○	○	○	○	○	○	○

(*) mandatory

Figure 5.12: GenderMag questionnaire: part 3.



GenderMag Questionnaire

Indicate your level of agreement with each of the statements listed below, selecting one per row (*)

	strongly disagree 1	2	3	4	neutral 5	6	7	8	strongly agree 9
I always do extensive research and comparison shopping before making important purchases	○	○	○	○	○	○	○	○	○
When a decision needs to be made, it is important to me to gather relevant details before deciding, in order to be sure of the direction we are heading	○	○	○	○	○	○	○	○	○
I avoid "advanced" buttons or sections in technology	○	○	○	○	○	○	○	○	○
I avoid activities that are dangerous and risky	○	○	○	○	○	○	○	○	○
Despite the risks, I use features in technology that haven't been proven to work	○	○	○	○	○	○	○	○	○

(*) mandatory

Figure 5.13: GenderMag questionnaire: part 4.

given a problem description and a new requirement. In the **understanding task**, participants had to answer questions about a given *[artefact]*. In the **reviewing task**, participants had to identify semantic defects on a given *[artefact]*, but we only informed the participants that their task was to find "defects". Explicitly describing the type of defects would have introduced a bias in the participants attention. This way, each participant was free to review the model using his best judgment, as a real-world stakeholder would. Typically, requirements modelling tools should protect the user against syntactic defects, hence our choice for semantic ones. The distribution of the tasks to the participants was random, but we balanced the number of participants performing each task. In all the tasks, the domain was a booking management system for an hotel.

Further details on the tasks are available in the specific Chapters 6 and 7.

5.1.5 Hypotheses, Parameters, and Variables

For each one of the high level goals, presented in Subsection 5.1.1, we define the **null** (H_0) and **alternative hypotheses** (H_1). The former states that there are **no** real underlying trends or patterns in the experiment setting and that the only reasons for differences in the observations are coincidental. We want to reject the null hypotheses with as high significance as possible. The latter states **there is** a statistically significant difference, and they are the hypotheses in favour of which the null hypotheses are rejected [245].

All the hypotheses are described in the specific Chapters 6, 7 and 8. However, for illustration purposes, we define the hypotheses for *i** versions, concerning *creation* tasks, which can be further refined to cope with *accuracy*, *speed*, *ease*, and *perceived effort*:

$\boldsymbol{H_{0N1}}$ Differences in the *i** version **do not** influence the *creation* of *i** SR models.

- $\boldsymbol{H_{0N1.1}}$ Differences in the *i** version **do not** influence the *accuracy to create i** SR models.
- $\boldsymbol{H_{0N1.2}}$ Differences in the *i** version **do not** influence the *speed to create i** SR models.
- $\boldsymbol{H_{0N1.3}}$ Differences in the *i** version **do not** influence the *ease to create i** SR models.

$\boldsymbol{H_{1N1}}$ Differences in the *i** version influence the *creation* of *i** SR models.

- $\boldsymbol{H_{1N1.1}}$ Differences in the *i** version influence the *accuracy to create i** SR models.
- $\boldsymbol{H_{1N1.2}}$ Differences in the *i** version influence the *speed to create i** SR models.
- $\boldsymbol{H_{1N1.3}}$ Differences in the *i** version influence the *ease to create i** SR models.

We follow the same approach to define the null and the alternative hypotheses for all the tasks and artefacts in every goal described in Section 5.1.1.

In Table 5.1, we present an overview of the **independent variables**. The first column presents the name of variable. The second column has the scale type, while the last

column has the options for the values, that is, the definition of each scale point. For *i* versions*, the variable is the *version*, which may be *i** 1.0, or iStar 2.0. For *use cases*, the variable is the *template*, which may be ARNE or ALCO. For *GenderMag*, the variable is the level of the facet – the *persona* – which may be Abby or Tim, on each of the 5 (five) facets (*motivation for using software*, *information processing style*, *computer self-efficacy*, *attitude towards risk*, and *ways of learning new technology*).

Table 5.1: Overview of the *independent* variables.

Name	Scale	Values
*i** version	nominal	{*i** 1.0; iStar 2.0}
Use case template	nominal	{ARNE; ALCO}
GenderMag persona	nominal	{Abby; Tim}

The **dependent variables** are *accuracy*, *speed*, *ease* (*visual*, *mental*, and *emotional*), and *perceived effort*. For each of these variables, there is a set of metrics, which were fully described in Chapter 4. From Table 5.2 to 5.7, we present an overview of these metrics. The first column shows the name of the variable, while the second one has its abbreviation. The third column presents the scale type, and the last column has the counting rule or formula for the metric calculation.

In Table 5.2 we present the metrics for the **dependent variable accuracy**. Higher values of *precision*, *recall*, *f-measure*, and *completeness* support the claim of a better *accuracy*. On the other hand, higher values of *complexity* suggest a worse *accuracy*. The metrics *complexity* and *completeness* are only applied in the creation and modification tasks, since in the understanding and reviewing tasks, the artefacts are not changed by the participant. Furthermore, their are applied to *i** 1.0 and iStar 2.0, and not to use cases.

Table 5.2: Overview of the metrics for the *dependent* variable *accuracy*.

Name	Abbreviation	Scale	Range	Counting rule
Precision	–	ratio	$0 \leq x \leq 1$	$\frac{number\ of\ relevant\ elements\ retrieved}{total\ number\ of\ retrieved\ elements}$
Recall	–	ratio	$0 \leq x \leq 1$	$\frac{number\ of\ relevant\ elements\ retrieved}{total\ number\ of\ relevant\ elements}$
F-measure	–	ratio	$0 \leq x \leq 1$	$\frac{2 * (precision * recall)}{(precision + recall)}$
Complexity	–	ratio	see Section 4.2 for the full list of metrics	
Completeness	–	ratio	see Section 4.2 for the full list of metrics	

In Table 5.3 we present the metrics for the **dependent variable speed**. Lower values of these metrics correspond to better *speed*. While the overall *duration* addresses the time spent in the task, *first action*, *last action*, *first detection* and *last detection* provide a detailed picture of the moment when the participant starts and ends providing valid feedback. The metrics *first action* and *first detection* are similar, but applied to different tasks. The former is used in the creation and modification tasks, while the latter is applied in the understanding and reviewing tasks. The same is valid for *last action* and *last detection*. A

higher value for *processing duration* indicates that the participant stopped working on the task, but decided to revise it before finishing.

Table 5.3: Overview of the metrics for the *dependent* variable *speed*.

Name	Abbreviation	Scale	Range	Counting rule
Duration	–	ratio	$0 \leq x$	$completion\ time - start\ time$
First action	FirstAct	ratio	$0 \leq x$	$first\ action\ time - start\ time$
Last action	LastAct	ratio	$0 \leq x$	$last\ action\ time - start\ time$
First detection	FirstDet	ratio	$0 \leq x$	$first\ detection\ time - start\ time$
Last detection	LastDet	ratio	$0 \leq x$	$last\ detection\ time - start\ time$
Processing duration	ProcDur	ratio	$0 \leq x$	$duration - last\ (action \vee detection)$

In Table 5.4 we present the metrics for the **dependent variable visual ease**, collected with the eye-tracking device. A higher *number* and *duration* of *fixations* is associated with a higher visual attention in a given set of AOIs (in this case, relevant vs. irrelevant model elements). Regarding the *average duration of fixation*, a higher value indicates more time and attention devoted to AOIs, which is correlated with cognitive processes. A higher *number of saccades* can be associated with a higher visual effort, meaning the participant may be somewhat "lost", making a more erratic navigation. A higher number of *saccades to the key* can also be associated with difficulties with the modelling language.

Table 5.4: Overview of the metrics for the *dependent* variable *visual ease*: eye-tracking

Name	Abbreviation	Scale	Range	Counting rule
Fixation rate on relevant elements	FixRel	ratio	$0 \leq x$	$\frac{number\ of\ fixations\ on\ the\ relevant\ AOI}{number\ of\ fixations\ on\ the\ AOG}$
Fixation rate on irrelevant elements	FixIrrel	ratio	$0 \leq x$	$\frac{number\ of\ fixations\ on\ the\ irrelevant\ AOI}{number\ of\ fixations\ on\ the\ AOG}$
Average duration of relevant fixations	AvgDurRelFix	ratio	$0 \leq x$	$\frac{\Sigma\ duration\ of\ fixations\ on\ the\ relevant\ AOI}{number\ of\ fixations\ on\ the\ relevant\ AOI}$
Average duration of irrelevant fixations	AvgDurIrrelFix	ratio	$0 \leq x$	$\frac{\Sigma\ duration\ of\ fixations\ on\ the\ irrelevant\ AOI}{number\ of\ fixations\ on\ the\ irrelevant\ AOI}$
Total number of saccades	TotSac	ratio	$0 \leq x$	$\Sigma\ saccades$
Total number of saccades to the key	Sac2Key	ratio	$0 \leq x$	$\Sigma\ saccades\ to\ the\ key\ AOI$

In Table 5.5 we present the metrics for the **dependent variable mental ease**, collected with the EEG scanner. The values for *average attention*, *average mental workload* and *average familiarity*, are calculated based on specific frequency bands, often referred to as alpha, beta, gamma, delta and theta. A higher *average attention* indicates the participant is engaged in the task, and a higher *average mental workload* indicates effort while performing it. For *average familiarity*, a higher value is associated with memory accessing and lower effort while performing the task.

In Table 5.6 we present the metrics for the **dependent variable emotional ease**, collected with the EDA scanner. A higher *average skin conductive level* is linked to a greater

Table 5.5: Overview of the metrics for the *dependent* variable *mental ease*: EEG

Name	Abbreviation	Scale	Range	Counting rule
Average attention	AvgAttention	ratio	$0 \leq x \leq 1$	$\frac{\Sigma\ attention\ value\ per\ ms}{total\ duration\ in\ ms}$
Average mental workload	AvgMentWL	ratio	$0 \leq x \leq 1$	$\frac{\Sigma\ mental\ workload\ value\ per\ ms}{total\ duration\ in\ ms}$
Average familiarity	AvgFam	ratio	$0 \leq x \leq 1$	$\frac{\Sigma\ familiarity\ value\ per\ ms}{total\ duration\ in\ ms}$

cognitive load, task difficulty, and stress. For computing the *heart rate variability*, we used features that represent the difference in time between two heart beats: RMSSD (root mean square of successive differences), and NN50 (the number of pairs of successive beat-to-beat intervals that differ more than 50ms). An increase in the *heart rate*, when in a stationary state, can be related with anxiety and mental stress.

Table 5.6: Overview of the metrics for the *dependent* variable *emotional ease*: EDA

Name	Abbreviation	Scale	Range	Counting rule
Average skin conductive level	AvgSCL	ratio	$0.2 \leq x \leq 4$	$\frac{\Sigma\ SCL\ in\ \mu s}{total\ number\ of\ SCL}$
Average RMSSD	AvgRMSSD	ratio	$10 \leq x \leq 120$	$\frac{\Sigma\ RMSSD\ in\ ms}{total\ number\ of\ RMSSD}$
Average NN50	AvgNN50	ratio	$1 \leq x \leq 70$	$\frac{\Sigma\ NN50\ in\ ms}{total\ number\ of\ NN50}$

In Table 5.7 we present the metrics for the **dependent variable perceived effort**. Higher values, in all the metrics, correspond to a greater perceived effort by the participant. Each metric is weighted, in terms of its importance for the overall effort. The denominator *15* corresponds to the 15 paired comparison of all the 6 dimensions to access the perceived workload.

Table 5.7: Overview of the metrics for the *dependent* variable *perceived effort* [227].

Name	Abbreviation	Scale	Range	Counting rule
Mental demand	MD	ratio	$0 \leq x \leq 100$	$\frac{mental\ rating * mental\ weight}{15}$
Physical demand	PD	ratio	$0 \leq x \leq 100$	$\frac{physical\ rating * physical\ weight}{15}$
Temporal demand	TD	ratio	$0 \leq x \leq 100$	$\frac{temporal\ rating * temporal\ weight}{15}$
Performance	Perf	ratio	$0 \leq x \leq 100$	$\frac{performance\ rating * performance\ weight}{15}$
Effort	Eff	ratio	$0 \leq x \leq 100$	$\frac{effort\ rating * effort\ weight}{15}$
Frustration	Frust	ratio	$0 \leq x \leq 100$	$\frac{frustration\ rating * frustration\ weight}{15}$
NASA-TLX Score	–	ratio	$0 \leq x \leq 100$	$MD + PD + TP + Eff + Perf + Frust$

5.1.6 Experimental Design

These studies follow a **quasi-experimental design**, since the allocation of participants to tasks was random, but without a pre-selection process. If a participant performed the creation task on an *i** 1.0 model, the next participant would be allocated to the modification task, so that the number of participants performing each task would be balanced. The exception to this allocation was in the iStar 2.0 creation and modification tasks, where we had 10 more participants than on the other tasks. This difference was caused by the way tasks were being allocated to participants in the beginning of the studies: instead of the previously described process, the participants were allocated to the same task until we had a reasonable number of participants performing that task. In both processes, in terms of tasks distribution, we have a **between subjects design**. This type of design is also called an independent measures design because every participant is only subjected to a single treatment, that is, only performs one of the tasks.

However, when our independent variables are the levels on each of the five GenderMag facets, and since we evaluate the differences in the levels of each facet for each participant, we have a **within subjects design**.

In a within subjects design, we can have a smaller number of participants, as every participant performs more than one task. Furthermore, there is a reduced variability due to subject differences. However, a learning effect from one task to the next could represent a confounding factor. Even if the order of the tasks was changed, the results may still be affected by the ordering.

With a between subjects design, on the other hand, there is no learning effect, nor a side effect due to ordering of the tasks. Nevertheless, it requires a higher number of participants and augments the variability due to subject differences. In order to reduce the latter, we have performed a random allocation of participants to tasks.

We opted for the between subjects design, in terms of tasks, for 3 (three) main reasons: (*i*) time; (*ii*) fatigue; and (*iii*) the learning effect. In particular in the studies with practitioners, time is a decisive factor. A quasi-experiment with multiple tasks may increase the mortality of the participants, or discourage them from participating, in the first place. Moreover, in a long experiment, the participants may become tired. This could decrease their performance on the last tasks. Alternatively, the learning effect may cause them to improve their performance over the course of the studies. Moreover, a crossover design, were every participant is subjected to more than one treatment, is complex [245] and it is sometimes discouraged based on the risk of performing an incorrect analysis [234].

5.2 Execution

5.2.1 Preparation

We carried out the data collection with a laptop connected to an external 22 inch, wide screen, full HD monitor; a The Eye Tribe eye-tracker (see Section 2.4.3); a NeuroSky

MindWave EEG headset (see Section 2.4.4); a BioSignalsPlux Wristband with BITalino EDA scanner (see Section 2.4.5); and an external mouse and keyboard. We prepared the session on the laptop, and the participant had access to the external monitor, mouse and keyboard. Participants sat on a chair without wheels, to avoid movements that could jeopardise the eye-tracker data.

We prepared the room setting so that all participants had similar conditions. In our University, the same meeting room was used for all the studies. For the experiments performed in software companies, the same room, in each company, was allocated to the entire day. The room was only being used for the studies, and there was only one participant in each session.

For the studies at our University, we scheduled the sessions according to participants' availability, with one hour between studies, so that the next participant would not have to wait too long, nor that the previous participant felt there were time constraints. For the studies at software companies, the participants appeared when they had a break on their normal work flow, and if a session was not being performed.

5.2.2 Procedure

When a participant arrived, (s)he sat on a chair in front of the external monitor, and was informed that the experiment consisted in watching a tutorial on a requirements language, and performing a task based on a problem description. We further informed the participant that we would be recording the contents on the screen, tracking the eyes movement, and collecting information of mental effort and heart rate. These explanations where necessary so that the participant could comfortably use the biometric devices. Finally, we explained (s)he could quit at any moment, and that there was no time limit for performing the task. The quasi-experiment procedure is represented in Figure 5.14.

The evaluation started with the participant ① **reading the consent form**. After that, (s)he ② **equipped the biometrics devices**. The EDA wristband was placed on the participant's non-dominant wrist, after removing any watches or bracelets. The buckle of the wristband was adjusted by the participant to a comfortable position (without it being too loose or too tight). Before putting the EEG headset, participants with earrings were asked to remove them. A special care was taken for participants with long hair, so that it would not obstruct the ear clip (which acts as a ground and reference). Due to the sensibility of the forehead sensor, we helped the participant to remove any foundation (cosmetics) from the forehead. We also helped participants with hair bangs, so that nothing was obstructing the forehead sensor of the EEG headset. We helped the participant seating comfortably so that the eyes would be around 50cm away from the screen. The eye-tracker was placed below the screen, without blocking it. We adjusted the eye-tracker's angle to cope with differences among the participants height. We then used the EyeTribe calibration application, only accepting *good* or *excellent* calibrations (top levels of a 5 points ordinal scale) to proceed to the actual data collection.

Figure 5.14: Experimental procedure, followed in all the quasi-experiments: ① consent form; ② biometrics devices; ③ video of fish swimming; ④ video tutorial; ⑤ task; ⑥ NASA-TLX; ⑦ demographic questionnaire; ⑧ GenderMag questionnaire.

We then asked the participant to ③ **watch the video of fish swimming**, while wearing the biometric sensors, allowing us to normalise the captured biometric data.

After that, the participant ④ **watched the video tutorial** on the corresponding requirements language and then started ⑤ **performing the task**, which was related with the modelling language the participant saw the tutorial on. The audio was recorded so that the participant could follow a *think aloud* approach. For the creation and modification tasks, talking was not necessary, as the answer was being recorded on the screen. However, the participant needed to give the answers to the understanding and reviewing tasks out loud. In all the cases, no (bio)feedback was provided to the participant during the entire evaluation, to avoid an unnecessary validity threat.

When the participant felt the task was completed, (s)he ⑥ **answered the NASA-TLX questionnaire**. Finally, each participant ⑦ **answered a questionnaire about demographic information**, where we offered the possibility of leaving an e-mail for received the aggregated results of the study, and ⑧ completed the **GenderMag questionnaire**.

In the end, we thanked the participant for taking the time to be part of the evaluation, and answered any questions (s)he might have.

5.2.3 Deviations from the Plan

During the modification task with iStar 2.0, there was a technical problem with the recording of the EEG data, which lost the connection with the computer twice during the collection process of 1 participant. Although the time that the collection was not made was only 11 seconds, we decided to still exclude the EEG data for that participant. The same happened during the review tasks of use cases ARNE, during 14 seconds, resulting

in the exclusion of the EEG data for that participant.

During the creation task with *i** 1.0, there was a technical problem with the recording of the EDA data, which lost the connection with the computer one time during the collection process of 3 participants. We decided to exclude the EDA data for those participants.

During the understanding task on use cases, there was a problem with the eye-tracker, that only recorded the screen but not the screen coordinates during the collection process. We decided to exclude the eye-tracking data for that participant.

5.3 Analysis

5.3.1 Data Set Preparation

In each session, we recorded without pausing the video and audio. During the data collection process, we took special care not to disturb, or distract, our participants.

When the evaluation ended, we watched the video with the audio, and manually collected the times when the participant started and ended the tasks, as well as the first and last actions or detections. These data allow us to analyse the participants' **speed**.

Since the answers were given orally, a preparation of that data was also necessary. For the understanding tasks, we had a table with all the elements present in the model, one per column. When listening to the answers, elements that a participant described as being the correct ones were marked with 1, in a row dedicated to each participant. For the reviewing tasks, the procedure was the same, but when the answer was different from the expected, we added a column with that answer, if it was not already present. At the end, the table contained all the answers given by the participants, and their frequency. For illustration purposes, in Figure 5.15 we present a fragment of the table for one of the questions of the understanding task.

In the creation and modification tasks using *i**, the model creation tool (see Section 4.2.3) collected all the elements added or modified by the participant in a CSV file. We manually compared the target model(s) file with the solution modelled by the participant. In the creation and modification tasks using use cases, the procedure was the same as for the understanding and reviewing tasks. These data allow us to analyse the participants' **accuracy**.

Concerning the eye-tracking data, the main areas of the stimulus and its elements were mapped into pixel coordinates to determine which regions and elements the participants were looking at, and saved in a CSV file. This enabled tagging the eye-tracking data with the elements being gazed at any given moment, which was a necessary step for computing the eye-tracking metrics. The fixations and corresponding durations were saved in a different CSV file, in order to calculate the normalised fixation durations.These data allow us to analyse the participants' **visual ease**.

Regarding the EEG and EDA scanners, the tools collecting the data save them in a CSV file. Those files have the structure needed to perform the analysis on the participant's

	A	B	C	D	E	F	G
1	Subject	Personal information	Available dates	System is down	Incongruent request	Credit card information	Booking confirmation received
2	341	1	0	0	0	0	0
3	342	0	1	0	0	0	0
4	343	1	1	0	0	0	0
5	344	1	1	1	0	0	0
6	345	1	1	0	0	0	0
7	346	1	0	0	1	0	0
8	347	1	0	0	0	1	0
9	348	1	1	0	0	0	0
10	349	1	1	0	0	0	0
11	350	1	1	0	0	0	0
12	351	1	1	0	0	0	0
13	352	1	1	0	0	0	1
14	353	1	1	0	0	0	0
15	354	1	1	0	0	0	0
16	355	1	0	0	0	0	0
17	356	1	1	0	0	0	0
18	357	1	1	0	0	0	0
19	358	1	1	0	0	0	0
20	359	1	1	1	1	0	1
21	360	1	1	1	0	0	0
22	361	1	1	0	0	0	0
23	362	1	1	1	0	0	0
24	363	1	0	1	0	0	0
25	364	1	1	1	0	0	0
26	365	1	0	0	0	0	0
27	366	1	1	1	0	0	0
28	367	1	1	0	0	0	1
29	368	0	1	0	0	0	1
30	369	1	0	1	0	0	0
31	370	1	1	0	0	0	0
32	371	1	1	0	0	0	1
33	372	1	0	0	0	0	0
34	373	0	1	0	0	0	0

Figure 5.15: Fragment of the data preparation for the understanding task.

mental and emotional ease, without further preparing the data. Similarly, no additional preparation is needed to analyse the participants' **perceived effort**, with NASA-TLX, nor to **characterise the participants**, with demographic data and GenderMag.

5.3.2 Analysis Procedure

We started by collecting descriptive statistics on our variables, to get an overview of their distribution. For quantitative measurement, we collected the *mean*, *standard deviation*, *skewness* and *kurtosis*. For qualitative measurements, we performed a frequency analysis. This was complemented with *box plots*, *Q-Q plots*, and *kernel density plots*, to help with the visual analysis of the distributions. This was then complemented with *Welch* t-*tests*. A discussion on the benefits of using Welch *t*-test for comparing distributions to detect statistically significant differences in a robust way (as opposed to two samples *t*-test, or a non-parametric alternative to it, such as the Mann-Whitney U test) is in [121]. The descriptive statistics on our variables are detailed in the specific Chapters 6 and 7.

5.4 Threats To Validity

As is every experimental work, even when carefully planned, there are some threats to validity that can bias the conclusions. Our work is no exception. For the identification of

the threats to validity, we are following Wohlin et al.'s guidelines [245]. They are divided in internal, external, construct and conclusion validity.

5.4.1 Internal Validity

We used a combination of convenience and snowball sampling. This can cause a *selection* threat, since the participants tend to be more motivated to be part of the experiments, considering that their participation is entirely voluntary. However, we found no evidence of this in the results.

5.4.2 External Validity

Overall, our participants had little to no prior knowledge of *i** 1.0 or iStar 2.0. Although this made them representatives of stakeholders with low requirements engineering expertise, by having participants with a greater level of experience we could analyse the differences between these two profiles. For use cases, on the other hand, 76% of participants have used at least one use case template in the past. As such, there is a difference in knowledge that may produce confounding results. This *interaction of selection and treatment* threat is an effect of selecting participants by convenience sampling. However, there was no statistically significant difference between those who were knowledgeable and those who were not. As such, we are confident that this threat has not compromised the results. However, one way to further verify it, is by replicating the experiments with a more heterogeneous group in terms of experience with the models.

Furthermore, the *i** models used in the understanding and reviewing tasks were relatively small, with only 2 actors and 25 elements (with 11 inside each actor, and 3 dependums). The problem description on the creation and modification tasks was also simple, in order to produce a small model as well. The use cases were also elementary, without presenting the description of alternative scenarios, nor *includes* or *excludes* dependencies. These models may not be representative of the ones used in industry, thus introducing an *interaction of setting and treatment* threat. In the performed quasi-experiments, we could not use larger models since we were limited by the technical specifications of the eye-tracker device, such as constraints in the external monitor dimensions and in the participant distance to the eye-tracker. The fonts and symbols used had to be big enough for easy visualisation by all participants. As such, the tested models are fragments of larger ones. Notwithstanding, presenting only model fragments to focus the attention of the stakeholders is a common technique for improving communication with them. Even so, in a future replication, it is important to vary the complexity of these models, to assess whether there is a significant variation on the success and effort on the tasks as models become more complex.

Moreover, we only analysed one domain: a booking management system for an hotel. As previously stated, we opted for a relatively known domain in order to reduce the effect of the results being related with difficulties in understanding the domain itself, and not

due to the requirements languages that were under study. We are also aware that tacit knowledge may play an important role in the performance of the participants. However, our goal was to evaluate the requirements languages, thus reducing confounding effects was considered a priority.

Finally, all the problem descriptions and tasks were in English. However, our participants have Portuguese as their mother tongue. We decided to create all the materials in English so they could be used in independent replications by international researchers. However, a limited English proficiency could have impacted the results. Nevertheless, all the participants were at ease with the English language and we found no impact of this decision in the results obtained.

5.4.3 Construct Validity

In all the quasi-experiments, we showed a video tutorial about the requirement model that was going to be used in the tasks. As such, participants might have felt that they were being evaluated. This may have caused an *evaluation apprehension* threat, where participants try to look better and thus confound the results. To mitigate this threat, we have not informed them about what exactly was being tested, that is, their accuracy, speed and ease in the performed tasks.

5.4.4 Conclusion Validity

Although we have a significant high number of participants, higher than most sample sizes reported, in particular, in other eye-tracking experiments (see [193]), sample size is always a risk, as results may not apply to even larger populations. We encourage replications of the quasi-experiments with a larger group. However, the distribution of participants on the GenderMag facets was not balanced. The distribution of participants to tasks did not take into account their facets, which may have influenced the results. Future replications can ask participants to first reply to the GenderMag questionnaire, and then assign the tasks in a way that the facets are evenly distributed across them.

5.5 Replication Package

We developed a web-based replication package [232] (Figure 5.16) for the learnability and appropriateness recognisability of *i** 1.0, iStar 2.0, ARNE use cases, and ALCO use cases. It can easily be extended to facilitate the evaluation of other quality characteristics and different software artefacts, and it does not require any additional installation or computer software besides a web browser. The replication package was implemented using web development programming languages and frameworks, namely HTML, CSS, Bootstrap, Javascript and jQuery. It is is optimised for a 22 inch, full HD monitor, with a 1920x1080 resolution. It is also optimised for Google Chrome, in full screen mode (F11 or Fn+F11). All the elements on the screen needed to be comfortably readable by our

participants, hence the optimisation for those sizes. However, the experiments may not appear correctly on screens with a different size.

The replication package is divided into 3 (three) main areas: metrics, tools, and experiments. The **metrics** area (Figure 5.17) has all the metrics defined in Subsection 4.2.2, for easy access and reference. The **tools** area (Figure 5.18) has the tools presented in Section 4.2.3, which allow the creation of *i** 1.0 and iStar 2.0 models, and the automated collection of the metrics. In the **experiments** area (Figure 5.19), all the quasi-experiments performed in this dissertation, for the evaluation of *i** and use cases, are available for consultation. Moreover, they can be used for **conducting independent replications**. The experiments are further described in Chapter 5.

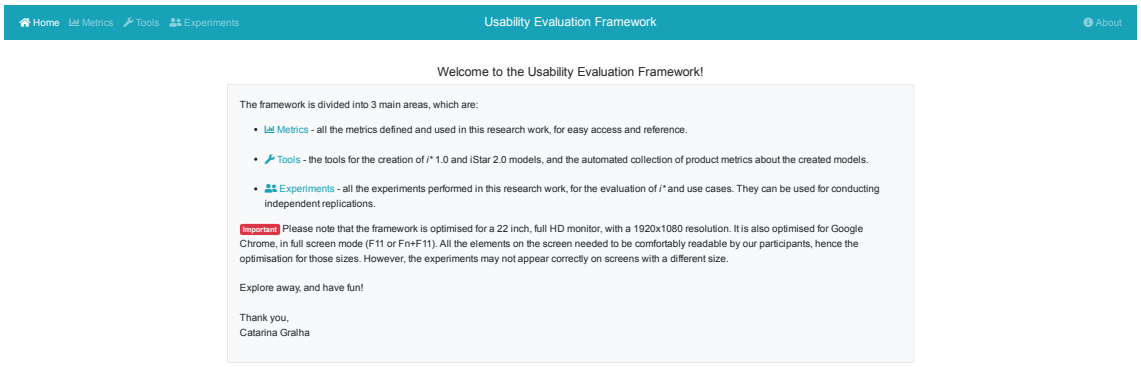


Figure 5.16: Homepage of the replication package.

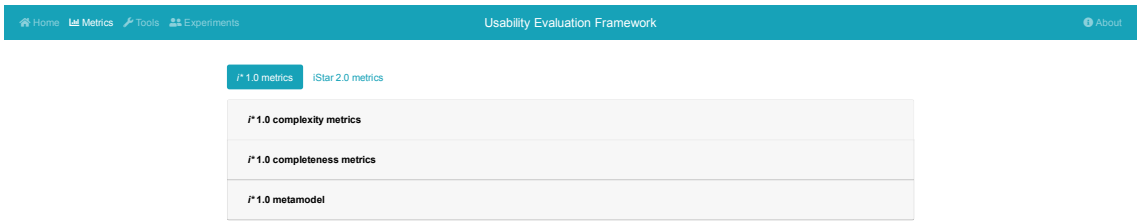


Figure 5.17: Metrics area of the replication package.

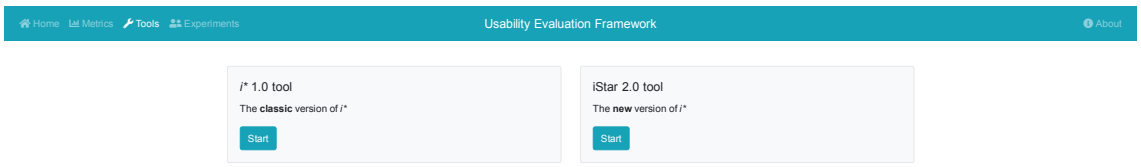


Figure 5.18: Tools area of the replication package.

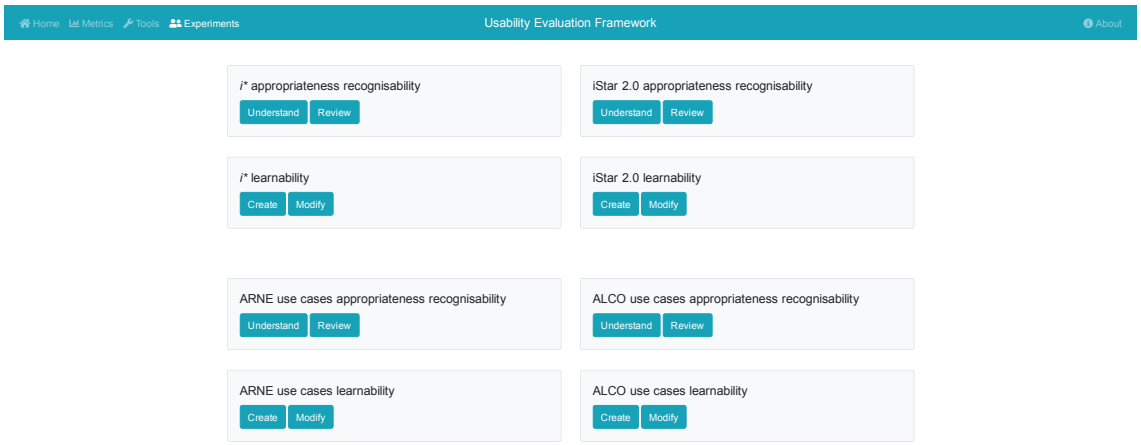


Figure 5.19: Experiments area of the replication package.

5.6 Summary

All the quasi-experiments performed in the context of this dissertation share a common methodology, fully detailed in this Chapter. The quasi-experiments were designed in a way they can be easily adapted to cover other requirements models or quality characteristics, by changing only 2 (two) steps of the experimental procedure: the video tutorial and the corresponding task. As such, this Chapter can be used as starting point for future replications of these studies, or as a basis for the planning and execution of similar experiments or quasi-experiments.

Chapter 6

Evaluation of *i** 1.0 and iStar 2.0

In this Chapter, we start by presenting the experimental protocol used in the *i** 1.0 and iStar 2.0 quasi-experiments, following Jedlitschka *et al.* guidelines [113] on how to report (quasi-)experiments in Software Engineering. The Chapter provides further details on the general experimental protocol previously presented in Chapter 5. It covers planning, execution, analysis, and discussion on the results and their implications, thus presenting a complete description of the empirical research performed for evaluating the appropriateness recognisability and learnability of *i** SR models. Although some Subsections are common to all the quasi-experiments, and fully described in Chapter 5, we decided to maintain the placeholders here, and make the reference to the corresponding Subsection on Chapter 5, for organisation purposes.

6.1 Experiments Planning

6.1.1 Goals

We describe our research goals using the GQM research goal template [7, 8]. We analyse differences in 2 (two) main sets, related with *i** versions, and levels of the GenderMag facets. Each set has 4 (four) main goals, each related with the tasks performed by the participants: creation, modification, understanding, and reviewing. Finally, each high level goal has a set of sub-goals, related with accuracy, speed, ease, and perceived effort, which are also defined. All the goals are similar, only changing the *underline and italic* part. However, they are fully specified for documentation purposes and easier reference.

The first set of goals is related with the ***i** versions* (**GN**) themselves. The objective is to compare the differences between the results achieved when using *i** 1.0 and iStar 2.0.

(GN1) Analyse differences in the *i** versions, **for the purpose of** evaluation, **with respect**

to their effects on the *creation* of *i** SR models, **from the viewpoint of** researchers, **in the context of** experiments conducted at our University and at software companies.

(GN1.1) Analyse differences in the *i** versions, **for the purpose of** evaluation, **with respect to** their effects on the *accuracy to create* *i** SR models, **from the viewpoint of** researchers, **in the context of** experiments conducted at our University and at software companies.

(GN1.2) Analyse differences in the *i** versions, **for the purpose of** evaluation, **with respect to** their effects on the *speed to create* *i** SR models, **from the viewpoint of** researchers, **in the context of** experiments conducted at our University and at software companies.

(GN1.3) Analyse differences in the *i** versions, **for the purpose of** evaluation, **with respect to** their effects on the *ease to create* *i** SR models, **from the viewpoint of** researchers, **in the context of** experiments conducted at our University and at software companies.

(GN2) Analyse differences in the *i** versions, **for the purpose of** evaluation, **with respect to** their effects on the *modification* of *i** SR models, **from the viewpoint of** researchers, **in the context of** experiments conducted at our University and at software companies.

(GN2.1) Analyse differences in the *i** versions, **for the purpose of** evaluation, **with respect to** their effects on the *accuracy to modify* *i** SR models, **from the viewpoint of** researchers, **in the context of** experiments conducted at our University and at software companies.

(GN2.2) Analyse differences in the *i** versions, **for the purpose of** evaluation, **with respect to** their effects on the *speed to modify* *i** SR models, **from the viewpoint of** researchers, **in the context of** experiments conducted at our University and at software companies.

(GN2.3) Analyse differences in the *i** versions, **for the purpose of** evaluation, **with respect to** their effects on the *ease to modify* *i** SR models, **from the viewpoint of** researchers, **in the context of** experiments conducted at our University and at software companies.

(GN3) Analyse differences in the *i** versions, **for the purpose of** evaluation, **with respect to** their effects on the *understanding* of *i** SR models, **from the viewpoint of** researchers, **in the context of** experiments conducted at our University and at software companies.

(GN3.1) Analyse differences in the *i** versions, **for the purpose of** evaluation, **with respect to** their effects on the *accuracy to understand* *i** SR models, **from the viewpoint of** researchers, **in the context of** experiments conducted at our University and at software companies.

(GN3.2) **Analyse** differences in the *i** versions, **for the purpose of** evaluation, **with respect to** their effects on the *speed to understand* *i** SR models, **from the viewpoint of** researchers, **in the context of** experiments conducted at our University and at software companies.

(GN3.3) **Analyse** differences in the *i** versions, **for the purpose of** evaluation, **with respect to** their effects on the *ease to understand* *i** SR models, **from the viewpoint of** researchers, **in the context of** experiments conducted at our University and at software companies.

(GN4) **Analyse** differences in the *i** versions, **for the purpose of** evaluation, **with respect to** their effects on the *reviewing* of *i** SR models, **from the viewpoint of** researchers, **in the context of** experiments conducted at our University and at software companies.

(GN4.1) **Analyse** differences in the *i** versions, **for the purpose of** evaluation, **with respect to** their effects on the *accuracy to review* *i** SR models, **from the viewpoint of** researchers, **in the context of** experiments conducted at our University and at software companies.

(GN4.2) **Analyse** differences in the *i** versions, **for the purpose of** evaluation, **with respect to** their effects on the *speed to review* *i** SR models, **from the viewpoint of** researchers, **in the context of** experiments conducted at our University and at software companies.

(GN4.3) **Analyse** differences in the *i** versions, **for the purpose of** evaluation, **with respect to** their effects on the *ease to review* *i** SR models, **from the viewpoint of** researchers, **in the context of** experiments conducted at our University and at software companies.

The second set of goals is related with the **levels of the GenderMag facets** (GGM). The objective is to compare the differences between the personas (Abby and Tim) on each of the 5 (five) facets (motivation for using software, information processing style, computer self-efficacy, attitude towards risk, and ways of learning new technology).

(GGM1) **Analyse** differences in the level of the GenderMag facets, **for the purpose of** evaluation, **with respect to** their effects on the *creation* of *i** SR models, **from the viewpoint of** researchers, **in the context of** experiments conducted at our University and at software companies.

(GGM1.1) **Analyse** differences in the level of the GenderMag facets, **for the purpose of** evaluation, **with respect to** their effects on the *accuracy to create* *i** SR models, **from the viewpoint of** researchers, **in the context of** experiments conducted at our University and at software companies.

(GGM1.2) **Analyse** differences in the level of the GenderMag facets, **for the purpose of** evaluation, **with respect to** their effects on the *speed to create* *i** SR models, **from the viewpoint of** researchers, **in the context of** experiments conducted at our University and at software companies.

(GGM1.3) **Analyse** differences in the level of the GenderMag facets, **for the purpose of** evaluation, **with respect to** their effects on the *ease to create* *i** SR models, **from the viewpoint of** researchers, **in the context of** experiments conducted at our University and at software companies.

(GGM2) **Analyse** differences in the level of the GenderMag facets, **for the purpose of** evaluation, **with respect to** their effects on the *modification* of *i** SR models, **from the viewpoint of** researchers, **in the context of** experiments conducted at our University and at software companies.

(GGM2.1) **Analyse** differences in the level of the GenderMag facets, **for the purpose of** evaluation, **with respect to** their effects on the *accuracy to modify* *i** SR models, **from the viewpoint of** researchers, **in the context of** experiments conducted at our University and at software companies.

(GGM2.2) **Analyse** differences in the level of the GenderMag facets, **for the purpose of** evaluation, **with respect to** their effects on the *speed to modify* *i** SR models, **from the viewpoint of** researchers, **in the context of** experiments conducted at our University and at software companies.

(GGM2.3) **Analyse** differences in the level of the GenderMag facets, **for the purpose of** evaluation, **with respect to** their effects on the *ease to modify* *i** SR models, **from the viewpoint of** researchers, **in the context of** experiments conducted at our University and at software companies.

(GGM4) **Analyse** differences in the level of the GenderMag facets, **for the purpose of** evaluation, **with respect to** their effects on the *reviewing* of *i** SR models, **from the viewpoint of** researchers, **in the context of** experiments conducted at our University and at software companies.

(GGM4.1) **Analyse** differences in the level of the GenderMag facets, **for the purpose of** evaluation, **with respect to** their effects on the *accuracy to review* *i** SR models, **from the viewpoint of** researchers, **in the context of** experiments conducted at our University and at software companies.

(GGM4.2) **Analyse** differences in the level of the GenderMag facets, **for the purpose of** evaluation, **with respect to** their effects on the *speed to review* *i** SR models, **from the viewpoint of** researchers, **in the context of** experiments conducted at our University and at software companies.

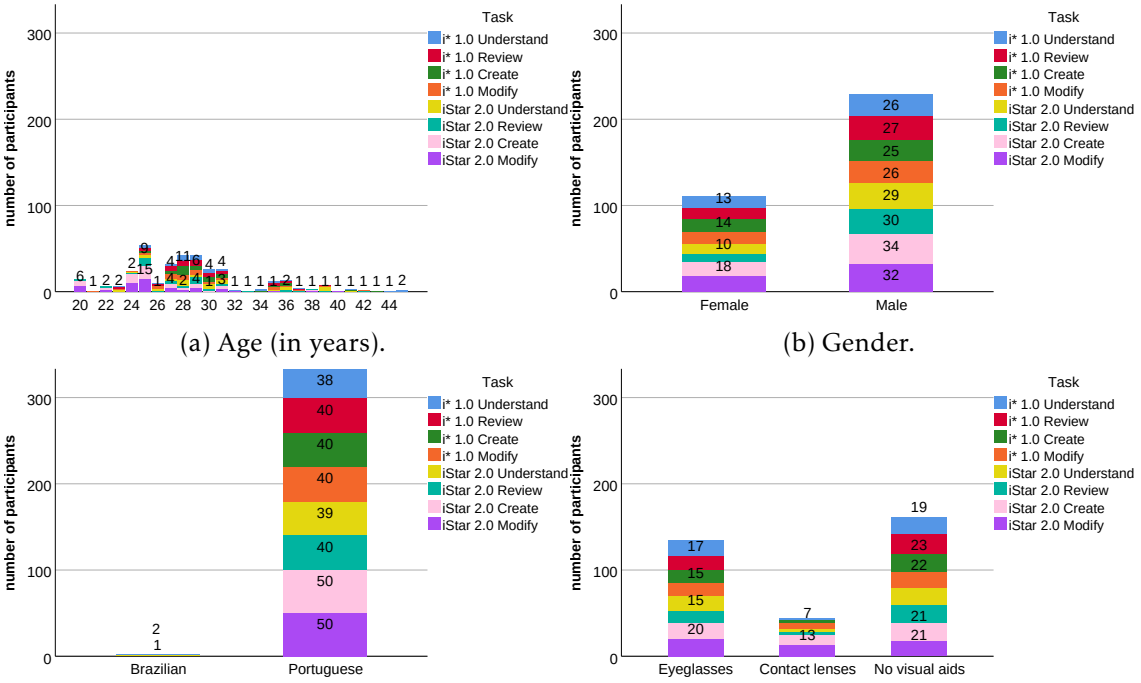
(GGM4.3) **Analyse** differences in the level of the GenderMag facets, **for the purpose of** evaluation, **with respect to** their effects on the *ease to review* *i** SR

models, **from the viewpoint of** researchers, **in the context of** experiments conducted at our University and at software companies.

6.1.2 Participants

We had 160 participants using *i** 1.0, and 180 using iStar 2.0, in a total of 340 participants. In the following bar charts, we present the number of participants per task, in a total of 8 (height) tasks: create, modify, understand and review for each of the *i** versions. The number of participants described in the text is the overall numbers for all the tasks.

Concerning participants *age* distribution (Figure 6.1a), they had between 20 and 45 years old, with an average of 28 years old. With respect to *gender* (Figure 6.1b), there were 229 male participants and 111 females. In terms of *nationality* (Figure 6.1c), 337 were Portuguese and 3 were Brazilian. Regarding the *usage of reading devices* (Figure 6.1d), 135 participants wore eyeglasses and 44 had contact lenses.



(a) Age (in years).

(b) Gender.

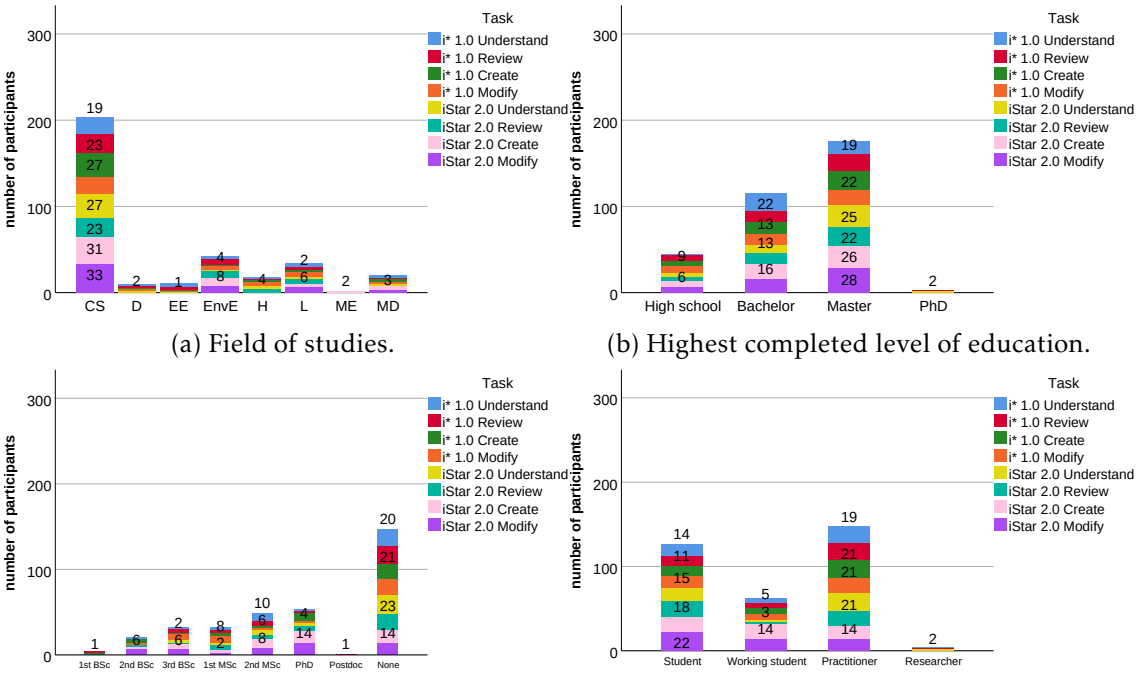
(c) Nationality.

(d) Usage of reading devices.

Figure 6.1: Participants general demographic information.

All participants had some university level training. Their *field of studies* (Figure 6.2a) spanned across multiple areas. We had 203 computer scientists (CS), 10 designers (D), 11 electrotechnical engineers (EE), 42 environmental engineers (EnvE), 18 historians (H), 34 lawyers (L), 2 mechanical engineers (ME), and 20 medical doctors (MD). For the *highest completed level of education* (Figure 6.2b), 45 completed high school, 116 concluded a BSc, 176 had a MSc, and 3 a PhD degree. Concerning *current level of education* (Figure 6.2c), 4 were in the first year of the BSc degree, 21 on the second year, and 32 on the third (and final) year. As for MSc students, 32 were in the first year, and 49 were on the second (and

final) year. Finally, 54 were doing a PhD, 1 was doing a Post-Doc, and 147 were no longer studying. The ones that were no longer studying had at least 4 years of experience. With respect to *current occupation* (Figure 6.2d), 127 of the participants were students, 64 were working students, 147 were practitioners, and 4 were researchers.



(a) Field of studies.

(b) Highest completed level of education.

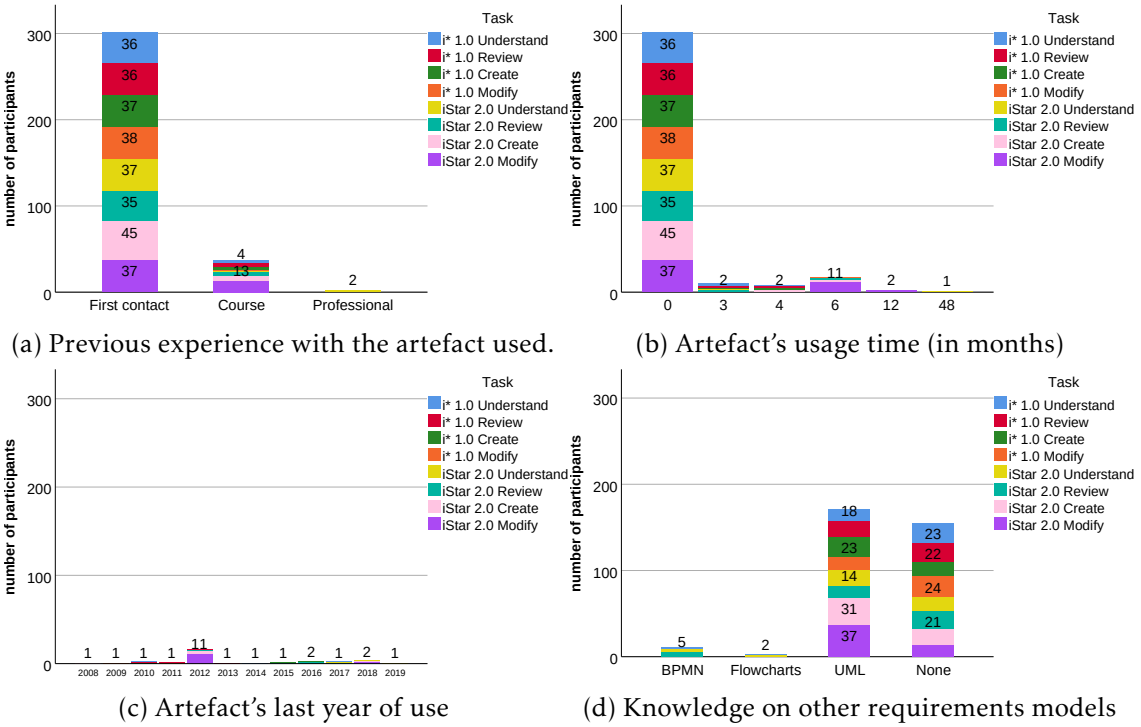
(c) Current level of education.

(d) Current occupation.

Figure 6.2: Participants academic and professional demographic information.

Regarding previous *experience* (Figure 6.3a) with the *i** version used in the task, for 301 participants it was their first contact with the version. However, 37 learnt it in the context of a course, and 2 in a professional environment. In those two latter scenarios, participants *usage time* with the versions (Figure 6.3b) had an average of 4 months. Participants tend to refer to the last usage time in terms of full years (for example, saying one or two years ago, and never one and a half years ago). On the other hand, some participants referred to 3, 4 or 6 months. We argue that all those months correspond to a University semester, depending on how people count. As for the *last use* of the version (Figure 6.3c), the vast majority of participants was no longer using it, and only had contact with *i** in a specific University course. Lastly, in terms of *knowledge on other requirements models* (Figure 6.3d), 171 participants claim to know UML in general, 11 referred to BPMN, and 3 specifically said to work with flowcharts in particular. The remaining 155 participants didn't report knowing any requirements language.

Participants spanned a reasonably wide range of values of each of the GenderMag facets, with 9 participants being characterised as a "pure" Abby and 24 as a "pure" Tim (Figures 6.4a and 6.4b). The other 307 participants had mixed characteristics of both Abby and Tim.



(a) Previous experience with the artefact used.

(b) Artefact's usage time (in months)

(c) Artefact's last year of use

(d) Knowledge on other requirements models

Figure 6.3: Participants knowledge on requirements models.

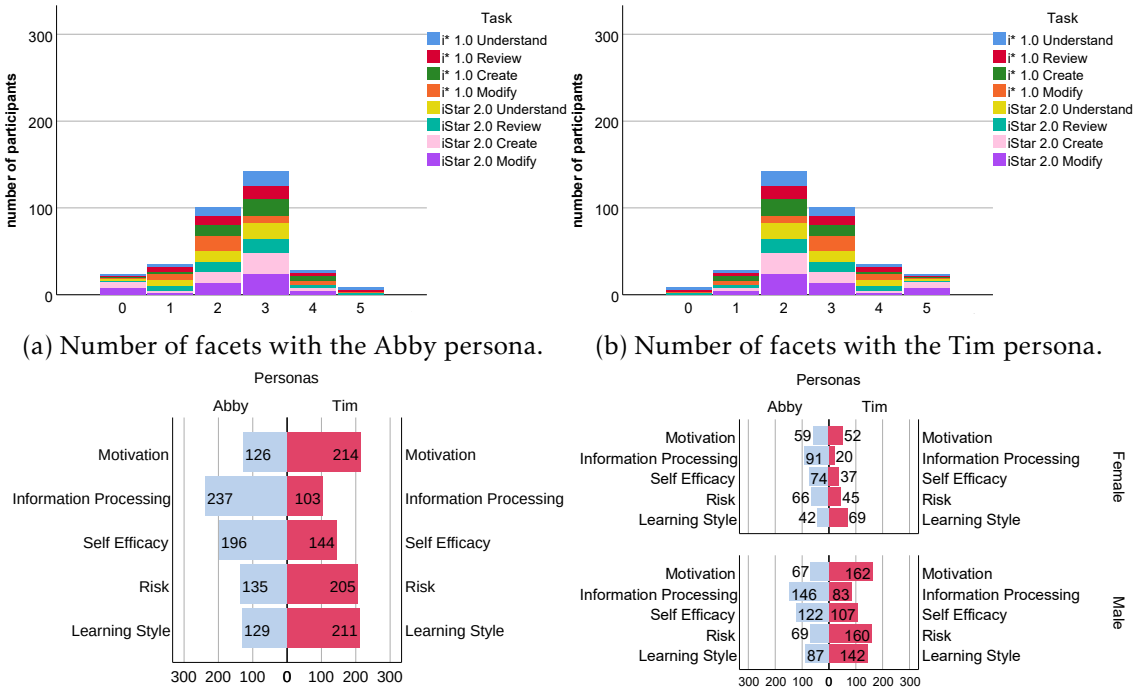
When analysing each facet (Figure 6.4c), the majority of the participants was identified as Tim in the *motivation*, *risk*, and *learning style* facets. For *information processing* and *self efficacy*, on the other hand, the majority of participants was described as Abby.

Taking a closer look into the relationship between the persona in each of the facets and the gender of participants (Figure 6.4d), the majority of female participants was characterised as Abby in all the facets, being *learning style* an exception. As for the males, the majority of participants was classified as Tim in all the facets, except for *information processing* and *self efficacy*. These results support the literature claim [10, 191] that characteristics in how people solve problems often cluster by gender.

6.1.3 Experimental Materials

The experimental materials included *(i)* a participant consent form; *(ii)* a video of fish swimming; *(iii)* a video tutorial about the *i** 1.0 or iStar 2.0; *(iv)* a problem description for the creation task; a problem description, an initial model and a new requirement for the modification task; a model and a set of questions for the understanding task; or a model with semantic defects for the review task; *(v)* a NASA-TLX questionnaire; *(vi)* a demographic questionnaire; and *(vii)* a GenderMag questionnaire. The materials *(i)*, *(ii)*, *(v)*, *(vi)* and *(vii)* were previously described in Section 5.1.3 of Chapter 5. The remainder materials, which are specific for the *i** versions, are described next.

The **video tutorial**, with 4 minutes for *i** 1.0 and 3 minutes and 35 seconds for iStar 2.0, explained the elements of an *i** 1.0 or iStar 2.0 model, depending on the artefact



(a) Number of facets with the Abby persona.

(b) Number of facets with the Tim persona.

(c) Personas in each facet.

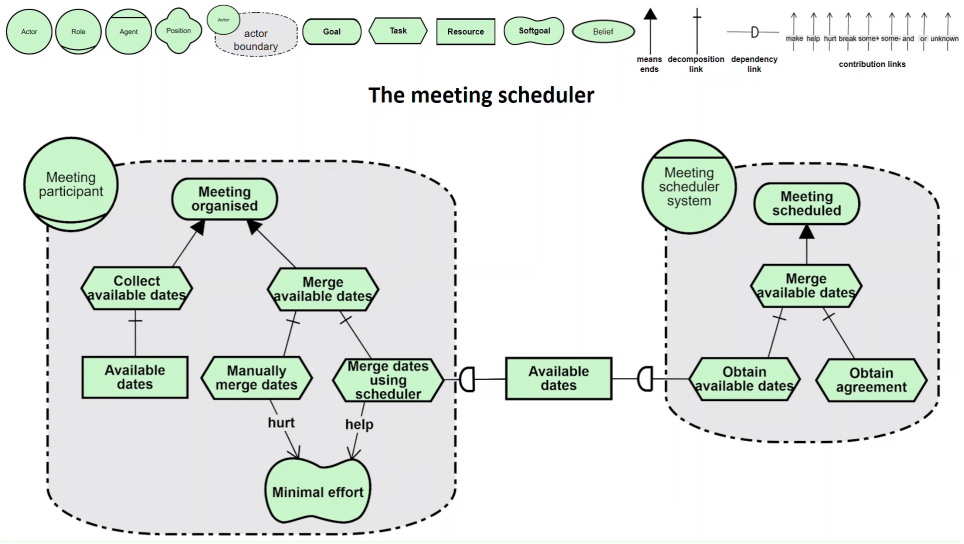
(d) Personas in each facet, by gender.

Figure 6.4: Participants distribution across GenderMag facets.

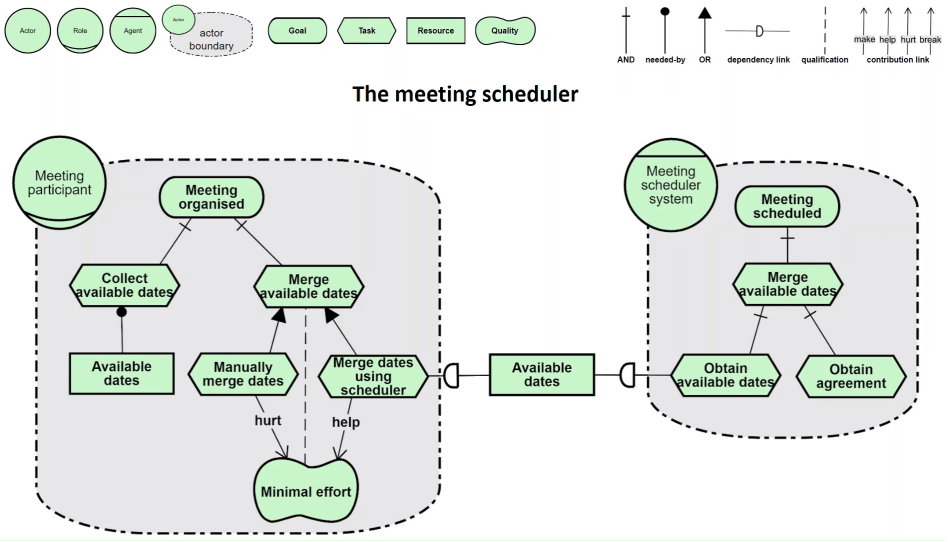
that was evaluated on that quasi-experiment. The tutorials have different durations, proportional to the number of elements and concepts in each *i** version. The tutorial included the construction of a correct model (similar to those that were going to be created, modified, understood or revised by the participants in the quasi-experiment) about a meeting scheduler system; and an audio and textual description of both the model elements, as they are being introduced, and their role in the model under construction. The modelling elements were described using the exact phrases and explanations present in the *i** 1.0 wiki guide [242], or in the iStar 2.0 Language Guide [49]. The participants had no control over the video, not being able to pause it or resume it, since having different viewing times and going through specifics parts of the tutorial more than one time could impact the results. A snapshot of the videos is presented in Figure 6.5a, for *i** 1.0; and in Figure 6.5b, for iStar 2.0.

In terms of **tasks**, we prepared 2 (two) versions of every material, one using *i** 1.0 and the other using iStar 2.0. The **creation** (Figure 6.6) and **modification** tasks (Figure 6.7), which are related with the **learnability** evaluation, share a common structure, with 3 (three) Areas Of Interest (AOI): the *problem description* on the left-hand side; the *editor's toolbar* on top; and the *canvas* where participants would create or modify the models.

The **understanding** (Figure 6.8) and **reviewing** (Figure 6.9) tasks, which are related with the **appropriateness recognisability** evaluation, share the same structure, with 3 (three) AOI: the *language key* on the left-hand side, the *question* the participant is suppose to answer on top, and the *i* model* about which the question is asked.



(a) Snapshot of the video tutorial for *i** 1.0.



(b) Snapshot of the video tutorial for iStar 2.0.

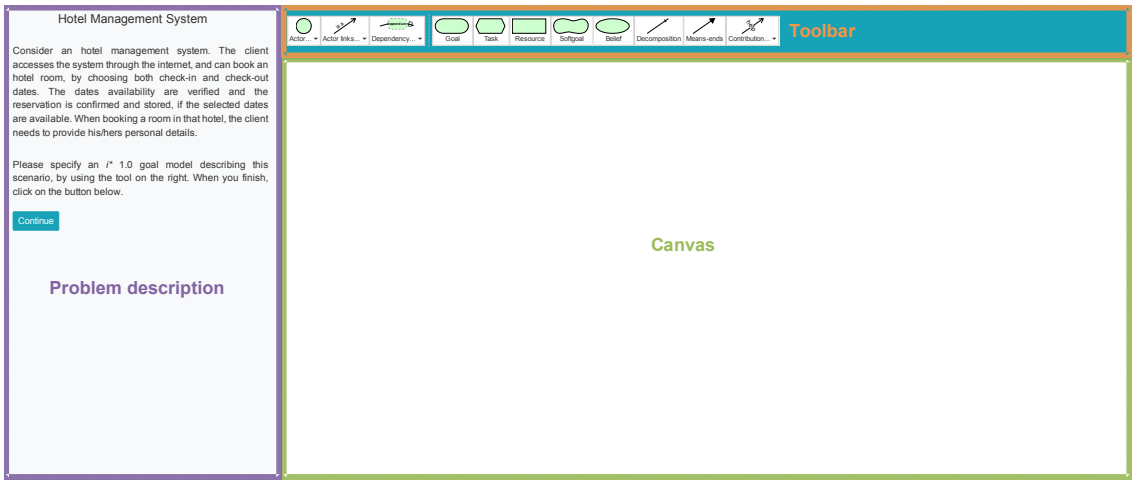
Figure 6.5: Snapshots of the *i** video tutorial viewed by the participants.

For each task, we used a similar layout with both versions, so that the only difference among them is the *i** version. For each task we further annotated 2 (two) sets of AOI to analyse eye-tracking data. An AOI is classified as *relevant* if it contains an element that belongs to the answer of the task, or *irrelevant* otherwise.

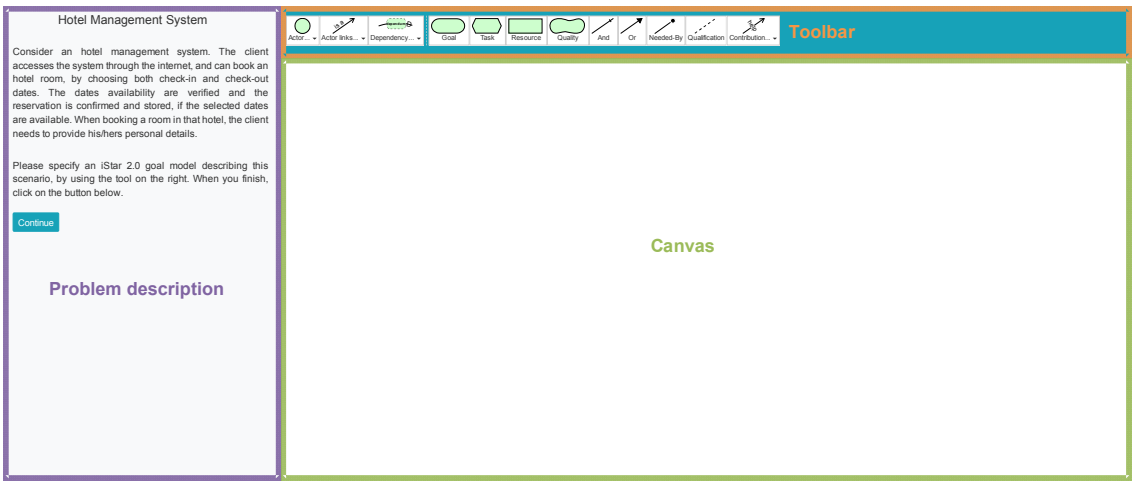
6.1.4 Tasks

For each *i** version, there were 4 tasks: creation, modification, understanding and review.

In the **creation task**, participants had to create an *i** model, given a small problem description, as we illustrated in Figure 6.6a, for *i** 1.0; and in Figure 6.6b for iStar 2.0. In Text 1, we present the problem description for both *i** versions.



(a) AOI for the creation task with *i** 1.0.



(b) AOI for the creation task with iStar 2.0.

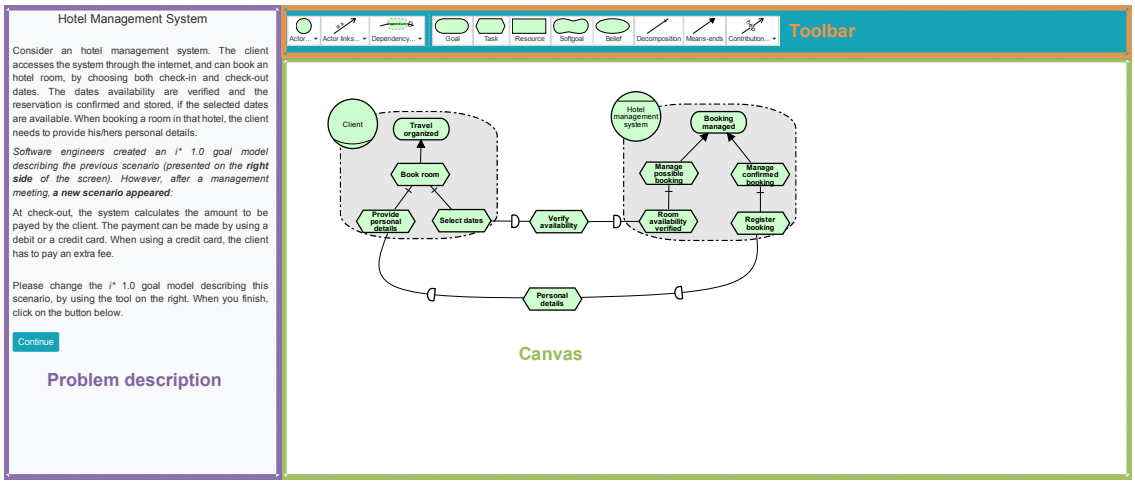
Figure 6.6: Creation task for *i**, illustrating the different AOI: the problem description on the left hand-side, the editor's toolbar on top, and the canvas on the remaining of the screen.

Text 1: Problem description for the *i creation task**

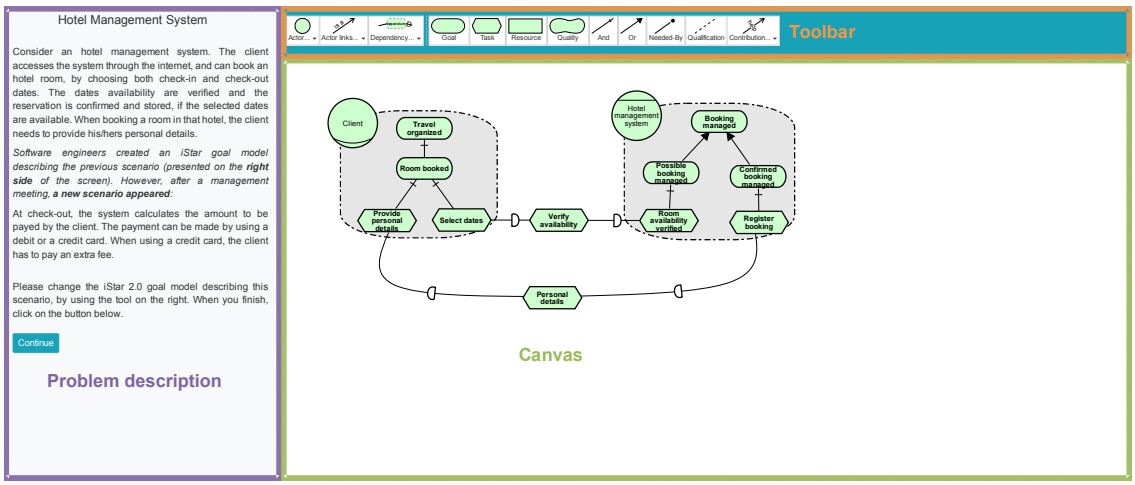
Consider an hotel management system. The client accesses the system through the internet, and can book an hotel room, by choosing both check-in and check-out dates. The dates availability are verified and the reservation is confirmed and stored, if the selected dates are available. When booking a room in that hotel, the client needs to provide his/her personal details.

Please specify an [i* 1.0/iStar 2.0] *goal model describing this scenario, by using the tool on the right. When you finish, click on the button below.*

In the **modification task**, participants had to modify an initial *i** model, given a



(a) AOI for the modification task with *i** 1.0.



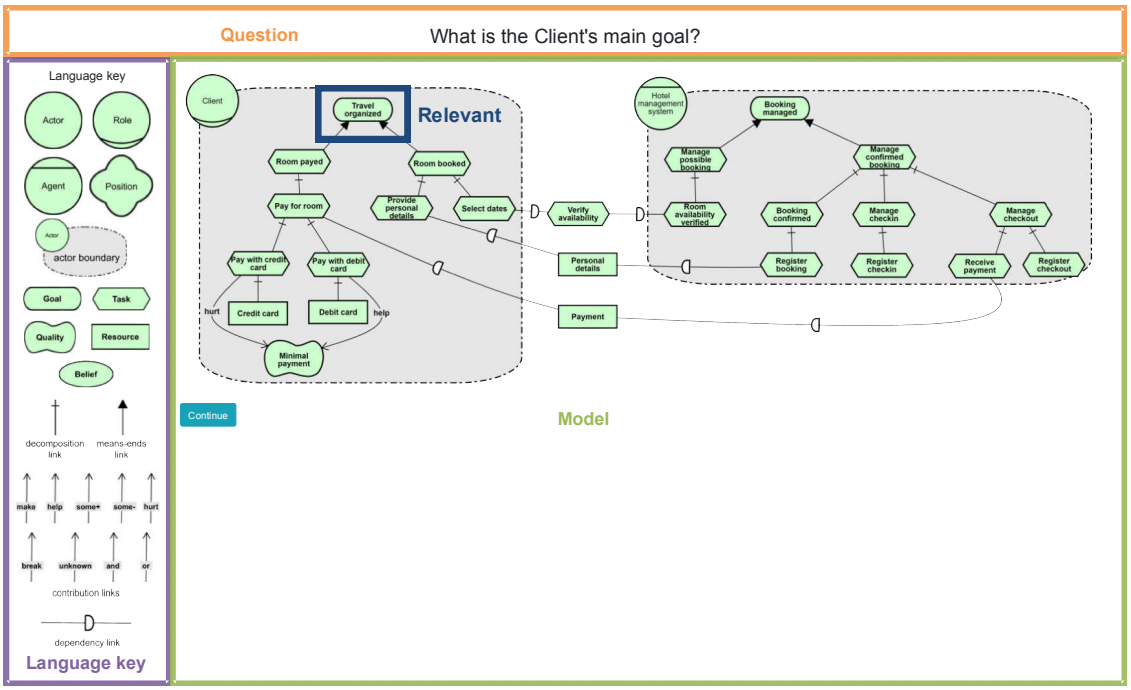
(b) AOI for the modification task with iStar 2.0.

Figure 6.7: Modification task for *i**, illustrating the different AOI: the problem description on the left hand-side, the editor's toolbar on top, and the canvas on the remaining of the screen, with the initial *i** SR model.

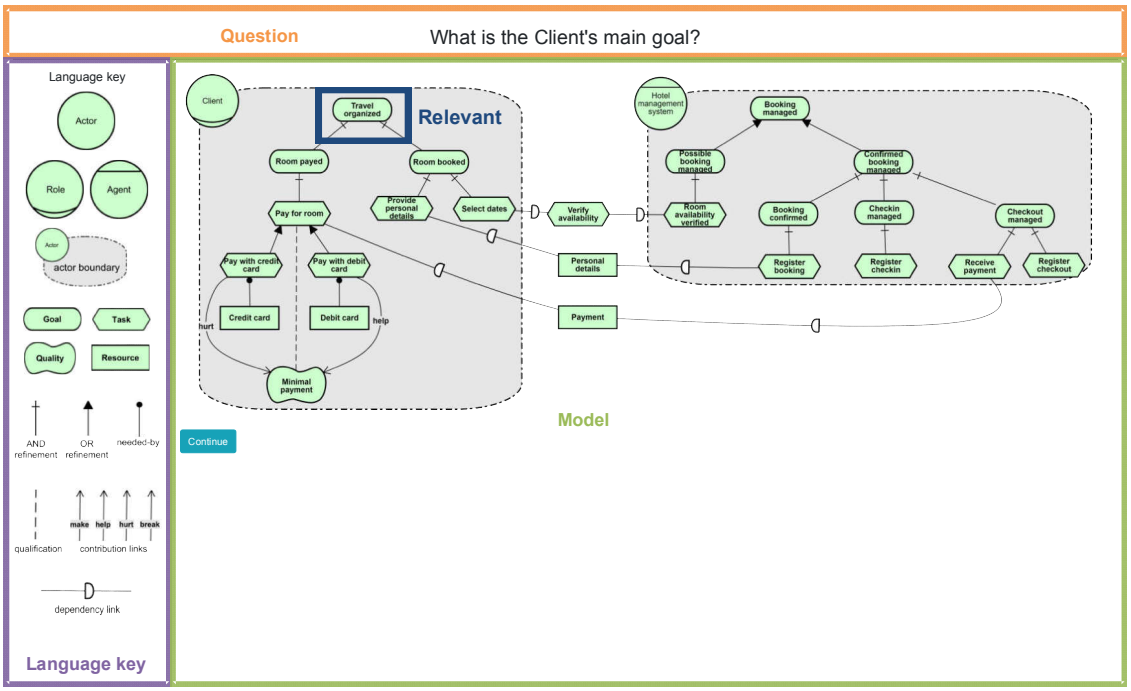
problem description and a new requirement, as we showed in Figure 6.7a, for *i** 1.0; and in Figure 6.7b, for iStar 2.0. In Text 2, we present the problem description and the new requirement, for both *i** versions.

Text 2: Problem description for the *i modification task**

Consider an hotel management system. The client accesses the system through the internet, and can book an hotel room, by choosing both check-in and check-out dates. The dates availability are verified and the reservation is confirmed and stored, if the selected dates are available. When booking a room in that hotel, the client needs to provide his/hers personal details.

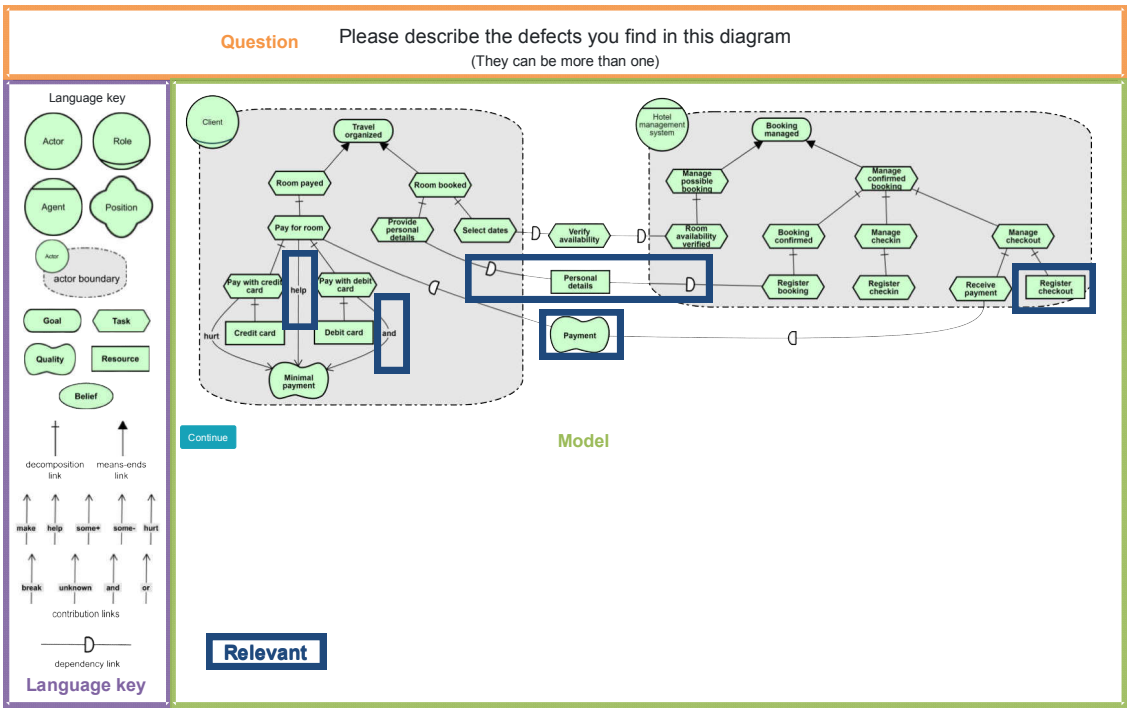


(a) AOI for the understanding task with *i** 1.0.

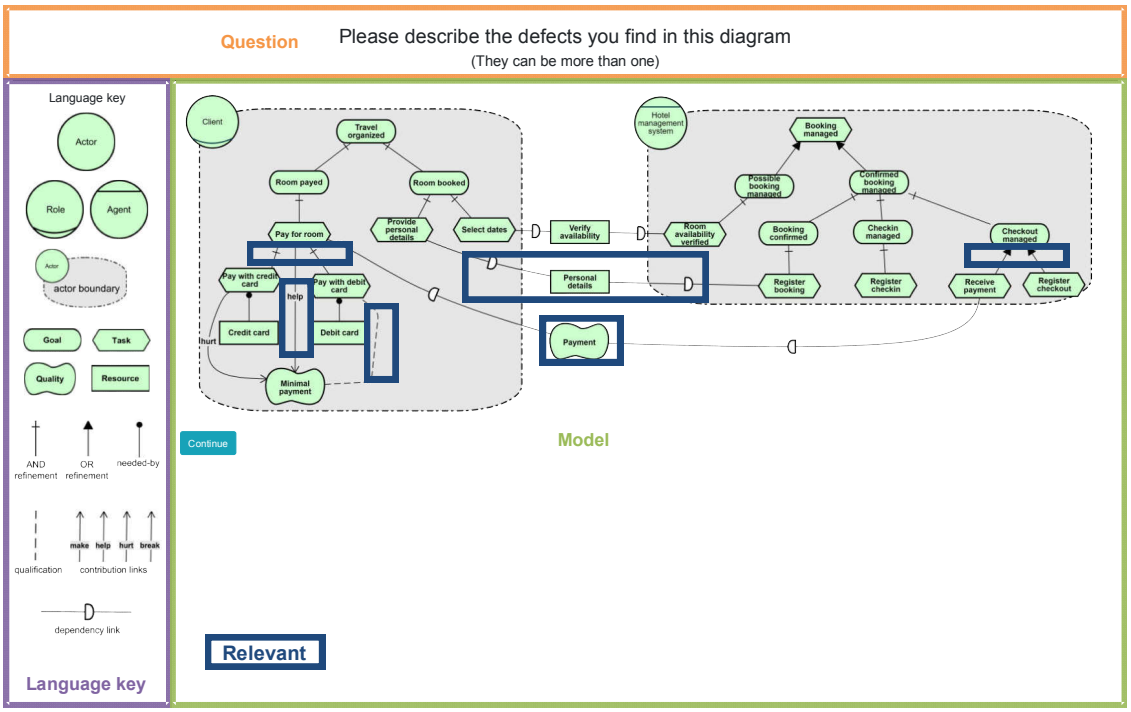


(b) AOI for the understanding task with iStar 2.0.

Figure 6.8: Understanding task for *i**, illustrating the different AOI: the question on top, the language key on the left-hand side, and the *i** model on the remaining of the screen.



(a) AOI for the review task with *i** 1.0.



(b) AOI for the review task with iStar 2.0.

Figure 6.9: Review task for *i**, illustrating the different AOI: the question on top, the language key on the left-hand side, and the *i** model on the remaining of the screen.

Software engineers created an [i* 1.0/iStar 2.0] *goal model describing the previous scenario (presented on the* ***right*** *side of the screen). However, after a management meeting,* ***a new scenario appeared****:*

At check-out, the system calculates the amount to be payed by the client. The payment can be made by using a debit or a credit card. When using a credit card, the client has to pay an extra fee.

Please change the [i* 1.0/iStar 2.0] *goal model describing this scenario, by using the tool on the right. When you finish, click on the button below.*

In the **understanding task**, participants had to answer a total of 7 (seven) questions about a given *i** SR model, as we presented in Figure 6.8a, for *i** 1.0; and in Figure 6.8b, for iStar 2.0. The questions, appearing in a random order, aimed to cover the main elements of an *i** SR model. In Text 3, we present the questions, in no particular order.

Text 3: Set of questions for the *i understanding task**

Which tasks are involved in making payments?
What is the Client's main goal?
What is the Hotel management system's main goal?
What is the best payment option for having a lower price?
Which tasks are involved in checking out?
Which resources may be needed to pay for a room?
On which resource is register a booking *dependent on?*

In the **reviewing task**, participants had to identify semantic defects on a given *i** SR model, as we illustrated in Figure 6.9a, for *i** 1.0; and in Figure 6.9b, for iStar 2.0. In Text 4, we present the assignment. We only informed the participants that their task was to find "defects". Explicitly describing the type of defects would have introduced a bias in the participants attention. This way, each participant was free to review the model using his best judgment, as a real-world stakeholder would. Typically, requirements modelling tools should protect the user against syntactic defects, hence our choice for semantic ones.

Text 4: Assignment for the *i reviewing task**

Please describe the defects you find in this diagram
(They can be more than one)

6.1.5 Hypotheses, Parameters, and Variables

For each one of the goals presented in Subsection 6.1.1, we define the **null** (H_0) and **alternative hypotheses** (H_1). Following the same principle of the goals, all the hypotheses

are similar, only changing the *underline and italic* part. However, they are fully specified for documentation purposes and easier reference.

The first set of hypotheses is related with the ***i* versions*** (H_{0Nx} and H_{1Nx}) themselves, with the objective of comparing the differences between the results achieved when using *i** 1.0 and iStar 2.0.

H_{0N1} Differences in the *i** version **do not** influence the *creation* of *i** SR models.

- $H_{0N1.1}$ Differences in the *i** version **do not** influence the *accuracy to create* *i** SR models.
- $H_{0N1.2}$ Differences in the *i** version **do not** influence the *speed to create* *i** SR models.
- $H_{0N1.3}$ Differences in the *i** version **do not** influence the *ease to create* *i** SR models.

H_{1N1} Differences in the *i** version influence the *creation* of *i** SR models.

- $H_{1N1.1}$ Differences in the *i** version influence the *accuracy to create* *i** SR models.
- $H_{1N1.2}$ Differences in the *i** version influence the *speed to create* *i** SR models.
- $H_{1N1.3}$ Differences in the *i** version influence the *ease to create* *i** SR models.

H_{0N2} Differences in the *i** version **do not** influence the *modification* of *i** SR models.

- $H_{0N2.1}$ Differences in the *i** version **do not** influence the *accuracy to modify* *i** SR models.
- $H_{0N2.2}$ Differences in the *i** version **do not** influence the *speed to modify* *i** SR models.
- $H_{0N2.3}$ Differences in the *i** version **do not** influence the *ease to modify* *i** SR models.

H_{1N2} Differences in the *i** version influence the *modification* of *i** SR models.

- $H_{1N2.1}$ Differences in the *i** version influence the *accuracy to modify* *i** SR models.
- $H_{1N2.2}$ Differences in the *i** version influence the *speed to modify* *i** SR models.
- $H_{1N2.3}$ Differences in the *i** version influence the *ease to modify* *i** SR models.

H_{0N3} Differences in the *i** version **do not** influence the *understanding* of *i** SR models.

- $H_{0N3.1}$ Differences in the *i** version **do not** influence the *accuracy to understand* *i** SR models.
- $H_{0N3.2}$ Differences in the *i** version **do not** influence the *speed to understand* *i** SR models.
- $H_{0N3.3}$ Differences in the *i** version **do not** influence the *ease to understand* *i** SR models.

H_{1N3} Differences in the *i** version influence the *understanding* of *i** SR models.

$H_{1N3.1}$ Differences in the *i** version influence the *accuracy to understand* *i** SR models.

$H_{1N3.2}$ Differences in the *i** version influence the *speed to understand* *i** SR models.

$H_{1N3.3}$ Differences in the *i** version influence the *ease to understand* *i** SR models.

H_{0N4} Differences in the *i** version **do not** influence the *reviewing* of *i** SR models.

$H_{0N4.1}$ Differences in the *i** version **do not** influence the *accuracy to review* *i** SR models.

$H_{0N4.2}$ Differences in the *i** version **do not** influence the *speed to review* *i** SR models.

$H_{0N4.3}$ Differences in the *i** version **do not** influence the *ease to review* *i** SR models.

H_{1N4} Differences in the *i** version influence the *reviewing* of *i** SR models.

$H_{1N4.1}$ Differences in the *i** version influence the *accuracy to review* *i** SR models.

$H_{1N4.2}$ Differences in the *i** version influence the *speed to review* *i** SR models.

$H_{1N4.3}$ Differences in the *i** version influence the *ease to review* *i** SR models.

The second set of hypotheses is related with the levels of the GenderMag facets (H_{0GMx} and H_{1GMx}), with the objective of comparing the differences between the personas on each of the 5 (five) problem-solving facets.

H_{0GM1} Differences in the level of the GenderMag facets **do not** influence the *creation* of *i** SR models.

$H_{0GM1.1}$ Differences in the level of the GenderMag facets **do not** influence the *accuracy to create* *i** SR models.

$H_{0GM1.2}$ Differences in the level of the GenderMag facets **do not** influence the *speed to create* *i** SR models.

$H_{0GM1.3}$ Differences in the level of the GenderMag facets **do not** influence the *ease to create* *i** SR models.

H_{1GM1} Differences in the level of the GenderMag facets influence the *creation* of *i** SR models.

$H_{1GM1.1}$ Differences in the level of the GenderMag facets influence the *accuracy to create* *i** SR models.

$H_{1GM1.2}$ Differences in the level of the GenderMag facets influence the *speed to create* i* SR models.

$H_{1GM1.3}$ Differences in the level of the GenderMag facets influence the *ease to create* i* SR models.

H_{0GM2} Differences in the level of the GenderMag facets **do not** influence the *modification* of i* SR models.

$H_{0GM2.1}$ Differences in the level of the GenderMag facets **do not** influence the *accuracy to modify* i* SR models.

$H_{0GM2.2}$ Differences in the level of the GenderMag facets **do not** influence the *speed to modify* i* SR models.

$H_{0GM2.3}$ Differences in the level of the GenderMag facets **do not** influence the *ease to modify* i* SR models.

H_{1GM2} Differences in the level of the GenderMag facets influence the *modification* of i* SR models.

$H_{1GM2.1}$ Differences in the level of the GenderMag facets influence the *accuracy to modify* i* SR models.

$H_{1GM2.2}$ Differences in the level of the GenderMag facets influence the *speed to modify* i* SR models.

$H_{1GM2.3}$ Differences in the level of the GenderMag facets influence the *ease to modify* i* SR models.

H_{0GM3} Differences in the level of the GenderMag facets **do not** influence the *understanding* of i* SR models.

$H_{0GM3.1}$ Differences in the level of the GenderMag facets **do not** influence the *accuracy to understand* i* SR models.

$H_{0GM3.2}$ Differences in the level of the GenderMag facets **do not** influence the *speed to understand* i* SR models.

$H_{0GM3.3}$ Differences in the level of the GenderMag facets **do not** influence the *ease to understand* i* SR models.

H_{1GM3} Differences in the level of the GenderMag facets influence the *understanding* of i* SR models.

$H_{1GM3.1}$ Differences in the level of the GenderMag facets influence the *accuracy to understand* i* SR models.

$H_{1GM3.2}$ Differences in the level of the GenderMag facets influence the *speed to understand* i* SR models.

$H_{1GM3.3}$ Differences in the level of the GenderMag facets influence the *ease to understand* *i** SR models.

H_{0GM4} Differences in the level of the GenderMag facets **do not** influence the *reviewing* of *i** SR models.

$H_{0GM4.1}$ Differences in the level of the GenderMag facets **do not** influence the *accuracy to review* *i** SR models.

$H_{0GM4.2}$ Differences in the level of the GenderMag facets **do not** influence the *speed to review* *i** SR models.

$H_{0GM4.3}$ Differences in the level of the GenderMag facets **do not** influence the *ease to review* *i** SR models.

H_{1GM4} Differences in the level of the GenderMag facets influence the *reviewing* of *i** SR models.

$H_{1GM4.1}$ Differences in the level of the GenderMag facets influence the *accuracy to review* *i** SR models.

$H_{1GM4.2}$ Differences in the level of the GenderMag facets influence the *speed to review* *i** SR models.

$H_{1GM4.3}$ Differences in the level of the GenderMag facets influence the *ease to review* *i** SR models.

For *i** *versions*, the **independent variable** is the *version*, which may be *i** 1.0, or iStar 2.0. For *GenderMag*, the variable is the level of the facet – the *persona* – which may be Abby or Tim, on each of the 5 (five) facets (motivation for using software, information processing style, computer self-efficacy, attitude towards risk, and ways of learning new technology).

The **dependent variables** are *accuracy*, *speed*, *ease* (*visual*, *mental*, and *emotional*), and *perceived effort*. The variables and the corresponding metrics were fully described in Subsection 5.1.5.

6.1.6 Experimental Design

See Subsection 5.1.6.

6.2 Execution

6.2.1 Preparation

See Subsection 5.2.1.

6.2.2 Procedure

See Subsection 5.2.2.

6.2.3 Deviations from the Plan

See Subsection 5.2.3.

6.3 Analysis

6.3.1 Data Set Preparation

See Subsection 5.3.1.

6.3.2 Analysis Procedure

See Subsection 5.3.2.

6.3.3 Descriptive Statistics

In Table 6.1 we present the descriptive statistics for the metrics collected in our data analysis. For the sake of brevity, we only present the results concerning *accuracy* of *i* versions*, and including *precision*, *recall*, and *f-measure*. Due to its high number, the remainder of the data can be found in a webpage [213].

For each metric, we present 8 lines in the Table. The first 2 refer to the creation task, the next 2 to the modification task, then 2 for the understanding task, and the last 2 are related with the review task. In the *Version* column we specify which of the *i** versions we are considering: *i** 1.0 or iStar 2.0. We further present the mean, standard deviation, skewness, kurtosis, and the p-value for the Shapiro-Wilk normality test. The shape of the distributions suggests that, in several cases, normality is **not** a reasonable assumption ($p < .05$). The variance of the distributions is not similar, for several of these variables.

The visual inspection of boxplot diagrams (in Figure 6.10), further reinforced our assessment concerning data normality.

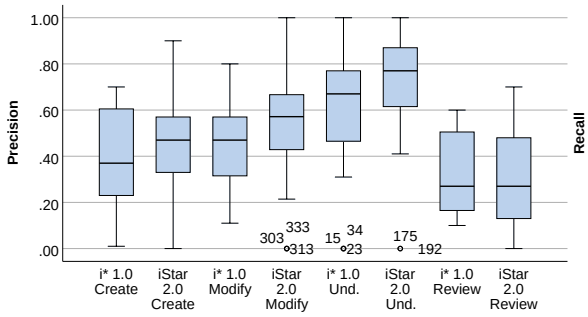
6.3.4 Hypotheses Testing

For testing our hypotheses, we used the *Welch's t-test*, as it is robust to deviations from the normal distribution, different sample sizes, and variance in the samples, thus following the recommendations on data analysis for Software Engineering empirical evaluations [121] (which summarises best practices in statistical analysis on other domains). We are using $p < .05$ for the level of significance and thus rejecting the null hypothesis.

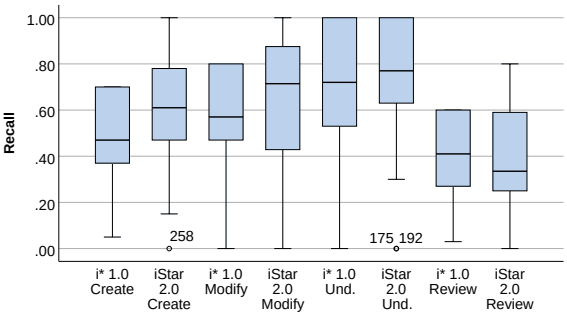
In HGM_x, related with the levels of the GenderMag facets, we are also interested in comparing the different levels with the versions, so we used the *Factorial ANOVA test*.

Table 6.1: Descriptive statistics for *accuracy* when using *i** 1.0 and iStar 2.0.

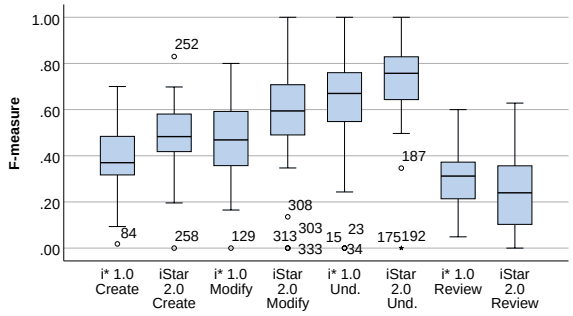
	Task	Version	Mean	S.D.	Skewness	Kurtosis	Shapiro-Wilk
Precision	Create	*i** 1.0	.409	.201	.242	-1.005	**.002**
		iStar 2.0	.494	.214	.211	-.351	.077
	Modify	*i** 1.0	.446	.201	.376	-.633	**.014**
		iStar 2.0	.571	.241	-.364	.474	.022
	Understand	*i** 1.0	.619	.267	-.562	.389	**.009**
		iStar 2.0	.711	.237	-1.236	2.331	**.001**
	Review	*i** 1.0	.330	.179	.473	-1.109	**.000**
		iStar 2.0	.290	.217	.517	-.885	**.002**
Recall	Create	*i** 1.0	.477	.207	-.359	-1.090	**.000**
		iStar 2.0	.611	.243	-.430	-.522	**.040**
	Modify	*i** 1.0	.559	.220	-.542	-.403	**.001**
		iStar 2.0	.664	.292	-.739	-.217	**.001**
	Understand	*i** 1.0	.670	.287	-1.033	.728	**.000**
		iStar 2.0	.759	.256	-1.436	2.136	**.000**
	Review	*i** 1.0	.410	.189	-.472	-1.143	**.000**
		iStar 2.0	.381	.241	.127	-1.082	**.013**
F-measure	Create	*i** 1.0	.394	.159	-.043	-.039	.607
		iStar 2.0	.486	.146	-.680	1.582	.117
	Modify	*i** 1.0	.462	.175	-.269	.186	.642
		iStar 2.0	.576	.218	-.887	1.593	**.003**
	Understand	*i** 1.0	.630	.235	-1.344	2.181	**.000**
		iStar 2.0	.714	.212	-1.951	5.065	**.000**
	Review	*i** 1.0	.321	.139	.305	-.369	.178
		iStar 2.0	.236	.148	.077	-.188	.057



(a) Precision for the *i** versions.



(b) Recall for the *i** versions.



(c) F-measure for the *i** versions.

Figure 6.10: Boxplots for *accuracy* when using *i** 1.0 and iStar 2.0.

For the sake of brevity, we only present the results concerning RQN1, which serve to illustrate the results for the hypothesis testing. Due to its high number, the remainder of the data can be found in a webpage [213]. In this Section, we only present the results for the hypotheses testing. The discussion on the data can be found in Section 6.4.

RQN1: Does a difference in the i* versions (i* 1.0 and iStar 2.0) influence the ability to* create *i* SR models?

In Table 6.2 we summarise the Welch t-test results for the creation task, when comparing i^* versions. There was a statistically significant difference in some of the variables ($p < .05$), with the p-value marked **bold** in the *Sig.* column of the Table.

Table 6.2: Welch t-test: *creation* task, i^* versions.

	Metric	Statistic	df1	df2	Sig. (p-value)
Accuracy	Precision	3.744	1	85.733	.056
	Recall	7.994	1	87.617	**.006**
	F-measure	7.930	1	80.148	**.006**
	Complexity	6.164	1	59.971	**.016**
	Completeness	17.244	1	86.749	**.000**
Speed	Duration	31.041	1	64.590	**.000**
	FirstAct	183.926	1	87.957	**.000**
	LastAct	3.920	1	61.855	.052
	ProcDur	57.540	1	49.365	**.000**
Visual ease	FixRel	579.196	1	84.540	**.000**
	FixIrrel	2.157	1	85.547	.146
	AvgDurRelFix	10.264	1	54.293	**.002**
	AvgDurIrrelFix	.005	1	72.031	.944
	TotSac	317.064	1	86.256	**.000**
	Sac2Key	874.131	1	87.338	**.000**
Mental ease	AvgAttention	.023	1	84.995	.879
	AvgMentWL	8.345	1	87.997	**.005**
	AvgFam	5.519	1	76.881	**.021**
Emot. ease	AvgSCL	22.067	1	85.992	**.000**
	AvgRMSSD	.688	1	81.232	.409
	AvgNN50	.602	1	85.502	.440
Perceived effort	Mental demand	.288	1	84.409	.593
	Physical demand	.000	1	83.497	1.000
	Temporal demand	.000	1	87.946	.987
	Effort	.001	1	85.437	.976
	Performance	29.356	1	83.356	**.000**
	Frustration	1.203	1	81.319	.276
	NASA-TLX Score	21.067	1	84.993	**.000**

Assessing accuracy. The *recall* achieved when using i^* 1.0 was *lower* ($M = .477$, $SD = .207$) than when using iStar 2.0 ($M = .611$, $SD = .243$, $t(1) = 7.994$, $p = .006$). Similarly, the *f-measure* achieved when using i^* 1.0 was *lower* ($M = .394$, $SD = .159$) than when using iStar 2.0 ($M = .486$, $SD = .146$, $t(1) = 7.930$, $p = .006$). The complexity of the

created *i** models when using *i** 1.0 was *higher* ($M = 52.200$, $SD = 24.116$) than when using iStar 2.0 ($M = 41.500$, $SD = 14.203\, t(1) = 6.164$, $p = .016$). Finally, the completeness of the created *i** models when using *i** 1.0 was *lower* ($M = 40.750$, $SD = 18.652$) than when using iStar 2.0 ($M = 58.020$, $SD = 20.735\, t(1) = 17.244$, $p = .000$).

Assessing speed. The overall *duration* of the task when using *i** 1.0 was *lower* ($M = 1236.425, SD = 238.842$) than when using iStar 2.0 ($M = 1790.660$, $SD = 650.759$, $t(1) = 31.041$, $p = .000$). However, the time for performing the *first action* when using *i** 1.0 was *higher* ($M = 715.500$, $SD = 151.796$) than when using iStar 2.0 ($M = 232.480$, $SD = 186.070$, $t(1) = 183.926$, $p = .000$). Finally, the *processing duration* when using *i** 1.0 was *lower* ($M = 19.000$, $SD = 17.328$) than when using iStar 2.0 ($M = 359.900$, $SD = 317.190$, $t(1) = 57.540$, $p = .000$).

Assessing visual ease. The *fixation rate on relevant elements* when using *i** 1.0 was *higher* ($M = 4.945$, $SD = .598$) than when using iStar 2.0 ($M = 1.861$, $SD = .612$, $t(1) = 579.196$, $p = .000$). On the other hand, the *average duration of relevant fixations* when using *i** 1.0 was *lower* ($M = 461.856$, $SD = 59.710$) than when using iStar 2.0 ($M = 594.800$, $SD = 285.722, t(1) = 10.264$, $p = .002$). The *total number of saccades* when using *i** 1.0 was *higher* ($M = 117.425$, $SD = 10.588$) than when using iStar 2.0 ($M = 75.840$, $SD = 11.515$, $t(1) = 317.064$, $p = .000$). Lastly, the *total number of saccades to the key* when using *i** 1.0 was also *higher* ($M = 99.075$, $SD = 9.515$) than when using iStar 2.0 ($M = 35.300$, $SD = 10.931$, $t(1) = 874.131$, $p = .000$).

Assessing mental ease. The *average mental workload* when using *i** 1.0 was *higher* ($M = .800$, $SD = .157$) than when using iStar 2.0 ($M = .692$, $SD = .198$, $t(1) = 8.345$, $p = .005$). Similarly, the *average familiarity* when using *i** 1.0 was *higher* ($M = .4750$, $SD = .169$) than when using iStar 2.0 ($M = .396$, $SD = .1442$, $t(1) = 5.519$, $p = .021$).

Assessing emotional ease. The *average skin conductive level* when using *i** 1.0 was *lower* ($M = 659.625$, $SD = 127.166$) than when using iStar 2.0 ($M = 790.720$, $SD = 136.845$, $t(1) = 22.067$, $p = .000$).

Assessing perceived effort. The perceived *performance* when using *i** 1.0 was *lower* ($M = 28.375, SD = 22.741$) than when using iStar 2.0 ($M = 62.500$, $SD = 36.565$, $t(1) = 29.356$, $p = .000$). In the same manner, the overall *NASA-TLX score* when using *i** 1.0 was *lower* ($M = 53.104$, $SD = 14.111$) than when using iStar 2.0 ($M = 70.420$, $SD = 21.5113$, $t(1) = 21.067$, $p = .000$).

6.4 Discussion

6.4.1 Evaluation of Results and Implications

RQN1: Does a difference in the i* versions (i* 1.0 and iStar 2.0) influence the ability to create i* SR models?

Assessing accuracy. Although the mean for the *precision* of participants using *i** 1.0

was lower than the ones using iStar 2.0, the difference in the distributions is not statistically significant. Nevertheless, the *precision* achieved by the participants was not great, being lower than 50% in both versions. Our interpretation is that participants struggled when creating *i** models, independently of the version used. There were some significant differences between the versions. Participants using *i** 1.0 had a lower *recall* and *f-measure* than the ones using iStar 2.0. Unlike the precision, the *recall* for iStar 2.0 was greater than 50%, which was not the case for *i** 1.0. The number of model elements available in *i** 1.0 version, in particular in terms of actor and contribution links, is greater than in iStar 2.0. This higher number of options may cause the participants to more easily select a link that is not appropriate for representing the relationship, and thus achieving a lower *recall*. The *complexity* of the models created with *i** 1.0 was higher, and the *completeness* was lower than with iStar 2.0. When further analysing the models and the corresponding complexity and completeness metrics, we note that the participants using *i** 1.0 tended to add more actors to the model, but without adding elements to the actor boundary. Similarly, in *i** 1.0 the number of *goals* and *softgoals* without decompositions was higher. Again in *i** 1.0, none of the participants used the specific actor *position*, the model element *belief*, nor the actor links *ins*, *occupies*, *covers* and *plays*. None of these model elements are available in iStar 2.0. For the problem description presented, those elements were not perceived as important. On the other hand, participants used all the model elements of the iStar 2.0 version, except for *qualifications*, which were never used. The *needed-by* link was only used in $\approx$ 20% of the created models. These links are new to the iStar 2.0 version, and using them was part of our solution. Their low usage suggests that participants were not able to perceive their importance. Further studies are needed to understand why.

Assessing speed. There was a statistically significant difference between the versions in terms of *duration*. Participants using *i** 1.0 were $\approx$ 10 minutes faster to complete the task than the ones using iStar 2.0. However, they started creating the model later on, taking more time to perform the *first action*, and to analyse the model elements available in the editor's toolbar, which can be observed with the eye-tracking data. The higher number of model elements available in *i** 1.0 may have caused this difference, since there are more elements to analyse and select. The *processing duration* was lower when using *i** 1.0, meaning that, after the creation of the model, the participant submits it without performing a thorough revision. We argue that this lack of a final analysis jeopardised the results and the creation of a more accurate model. As such, an overall *duration* being lower is not always positive and does not translate into a higher *accuracy*.

Assessing visual ease. There was a statistically significant difference between the versions in terms of visual ease. Participants using *i** 1.0 had a greater visual effort than the ones using iStar 2.0, observable through a higher *fixation rate on relevant elements*, *total number of saccades* and *total number of saccades to the key* (in this case, the editor's toolbar). However, the *average duration of relevant fixations* was lower. Our interpretation is that, although the participants were looking at the right elements, they were not able to identify them as relevant, rapidly changing to other model elements and thus making a

more erratic navigation. Once more, we argue that the higher number of model elements available in *i** 1.0 may have caused this difference, since there are more elements to analyse, and the participants might have felt somewhat "lost" with the amount of options to select from. Since participants created a wide range of different models, having an heat map representing the fixations and duration of those fixations would not provide us useful insights. As such, we decided not to create the heat map for this task.

Assessing mental ease. Although the mean for the *average attention* of participants using *i** 1.0 was slightly higher than in the ones using iStar 2.0, the difference in the distributions is not statistically significant. Nevertheless, the *average attention* was higher than 70% in both versions. Participants were mentally engaged and attentive to the task they were performing, independently of the version. There were some significant differences between the versions. The *average mental workload* was higher in participants using *i** 1.0 than in the ones using iStar 2.0, indicating a greater effort while performing the task. The *average familiarity* was lower than 50% for both versions. However, it was higher for participants using *i** 1.0. Some of the participants said to have learnt *i** 1.0 in the context of a course and worked with it during a University semester, hence an initial *familiarity* with the model elements. However, this greater *average familiarity* had no impact on the *accuracy* of the task, nor in the *mental effort* while performing it.

Assessing emotional ease. There was no statistically significant difference between the versions in terms of *heart rate variability*, for both RMSSD and NN50. However, there was one significant difference between the versions. The *average skin conductive level* of participants using *i** 1.0 was lower than the one of participants using iStar 2.0. We argue that, although their *mental workload* and *visual effort* was higher, participants were more relaxed and less stressed while performing the task. Our interpretation is that the participants were engaged in the task, but did not feel the pressure for achieving a high performance, nor an evaluation apprehension.

Assessing perceived effort. Although there was no statistically significant difference between the versions in terms of perceived *mental demand*, it was higher than 80 (out of 100) for both versions, which is in line with the biometric data. When further analysing the other components of NASA-TLX, we note that participants perceived the task of creating an *i** model, independently of the version, as being mentally challenging, strenuous and somewhat frustrating. This is in line with the results obtained through the biometric data. There were some significant differences between the versions. The perceived *performance* of participants using *i** 1.0 was lower than the one of participants using iStar 2.0. The overall *NASA-TLX score* was also lower for participants using *i** 1.0. This is congruent with the results obtained in terms of *accuracy*, meaning the participants were well aware of their *performance* on the task.

RQN2: Does a difference in the i* versions (i* 1.0 and iStar 2.0) influence the ability to* modify *i* SR models?

Assessing accuracy. There was a statistically significant difference between the versions in terms of *precision* and *f-measure*. Participants using *i** 1.0 had a lower *precision* and *f-measure* than the ones using iStar 2.0. Although the *precision* and *f-measure* achieved by the participants was not great with any of the versions, these values were higher than 50% for iStar 2.0, which was not the case for *i** 1.0. Our interpretation is that participants struggled when modifying *i** models, but they were able to achieve a better result when using iStar 2.0. The mean for the *recall* of participants using *i** 1.0 was lower than the ones using iStar 2.0, but the differences in the distributions is not statistically significant. However, these results further reinforce the notion that iStar 2.0 lead to a higher accuracy. We argue that, since the number of model elements is higher in *i** 1.0 than in iStar 2.0, this larger number of options may have caused to participants to be confused on which element to select. The *complexity* of the models created with *i** 1.0 was higher, and the *completeness* was lower than with iStar 2.0. When further analysing the models and the corresponding complexity and completeness metrics, we note that participants using *i** 1.0 tended to add more actors to the model, and change the initial model provided with the task, by removing model elements or changing their labels. The number of *goals* without decompositions was higher when using *i** 1.0, as well as actors with only one element inside. In terms of *contribution links*, participants had difficulties in understanding the differences among them, using the links indiscriminately. With iStar 2.0, however, the *contribution links* were correctly used in all the models. The number of *contribution links* types is higher in *i** 1.0, which may cause the participants to more easily select a link that is not appropriated for representing the relationship. Similarly to the creation task, in *i** 1.0 none of the participants used the specific actor *position*, the model element *belief*, nor the actor links *ins*, *occupies*, *covers* and *plays*. None of these model elements are available in iStar 2.0. For the problem description presented, those elements were not perceived as important. On the other hand, participants used all the model elements of the iStar 2.0 version, except for *qualifications*, which were never used. The *needed-by* link was used in $\approx$ 55% of the created models, which is higher than in creation task. These links are nre to the iStar 2.0 version, and using them was part of our solution. However, participants were not able to perceived the importance of the *qualification* link, and further studies are needed to understand why.

Assessing speed. Although the mean for the *duration* of participants using *i** 1.0 was lower than the ones using iStar 2.0, the difference in the distributions is not statistically significant. Actually, participants using *i** 1.0 were less than 2 minutes faster to complete the task than the ones using iStar 2.0. However, there were some significant differences between the versions. Participants using *i** 1.0 started creating the model later on, taking more time to perform the *first action*. Our interpretation is that the higher number of model elements available in *i** 1.0 may have caused this difference, since there are more elements to analyse and select. The time for performing the *last action* was also higher for participants using *i** 1.0 than the ones using iStar 2.0. The modification of an *i** 1.0 model takes longer, possibly due to the higher number of model elements to select

from. On the other hand, the *processing duration* was lower when using *i** 1.0, meaning that, after the modification of the model, the participant submits it without performing a thorough revision. We argue that this lack of final analysis jeopardised the results and the modification of a model into a more accurate one. As such, an overall *duration* being lower is not always positive and does not translate into a higher *accuracy*.

Assessing visual ease. There was a statistically significant difference between the versions in terms of visual ease. Participants using *i** 1.0 had a greater visual effort then the ones using iStar 2.0, observable through a higher *fixation rate on relevant elements*, *total number of saccades*, and *total number of saccades to the key*. However, the *average duration of relevant fixations* was lower. Our interpretation it that, although participants were looking at the right elements, they were not able to identify them as relevant, rapidly changing to other model elements and thus making a more erratic navigation. We argue that the higher number of model elements available in *i** 1.0 may have caused this difference, since there are more elements to analyse, and the participants might have felt somewhat "lost" with the amount of options to select from. Since participants modified the models in a wide range of different ways, having an heat map representing the fixations and duration of those fixations would not provide us useful insights. As such, we decided not to create the heat map for this task.

Assessing mental ease. The mean for the *average attention* and *average mental workload* is highly similar in both versions, hence the differences in the distributions not being statistically significant. Nevertheless, these metrics were higher than 65% in both versions. Participants were mentally engaged and attentive to the task they were performing, independently of the version. There was one significant difference between the versions. Participants using *i** 1.0 had a higher *average familiarity* than the ones using iStar 2.0. Some participants said to have learnt *i** 1.0 in the context of a course and worked with it during a University semester, hence the initial *familiarity* with the model elements. However, this greater *familiarity* had no impact on the *accuracy* of the task, nor on the *mental effort* while performing it.

Assessing emotional ease. There was a statistically significant difference between the versions in terms of emotional ease. Participants using *i** 1.0 had a lower emotional effort than the ones using iStar 2.0, observable through a lower *average skin conductive level* and *heart rate variability* (for RMSSS). We argue that, although their *visual effort* was higher, they were more relaxed and less stressed while performing the task. Our interpretation is that the participants have not felt the pressure for achieving a high performance, nor an evaluation apprehension.

Assessing perceived effort. Although the mean for the perceived *mental demand* of participants using *i** 1.0 was slightly higher than in the ones using iStar 2.0, the difference in the distribution is not statistically significant. Nevertheless, the perceived *mental demand* was higher than 75 (out of 100) for both versions. The participants perceived the task of modifying an *i** model, independently of the version, as being mentally challenging, which is in line with the results obtained through the biometric data. There were

significant differences between the versions. The perceived *performance* of participants using *i** 1.0 was lower than the one of participants using iStar 2.0. The overall *NASA-TLX score* was also lower for participants using *i** 1.0. This is congruent with the results obtained in terms of *accuracy*, meaning the participants were well aware of their *performance* on the task.

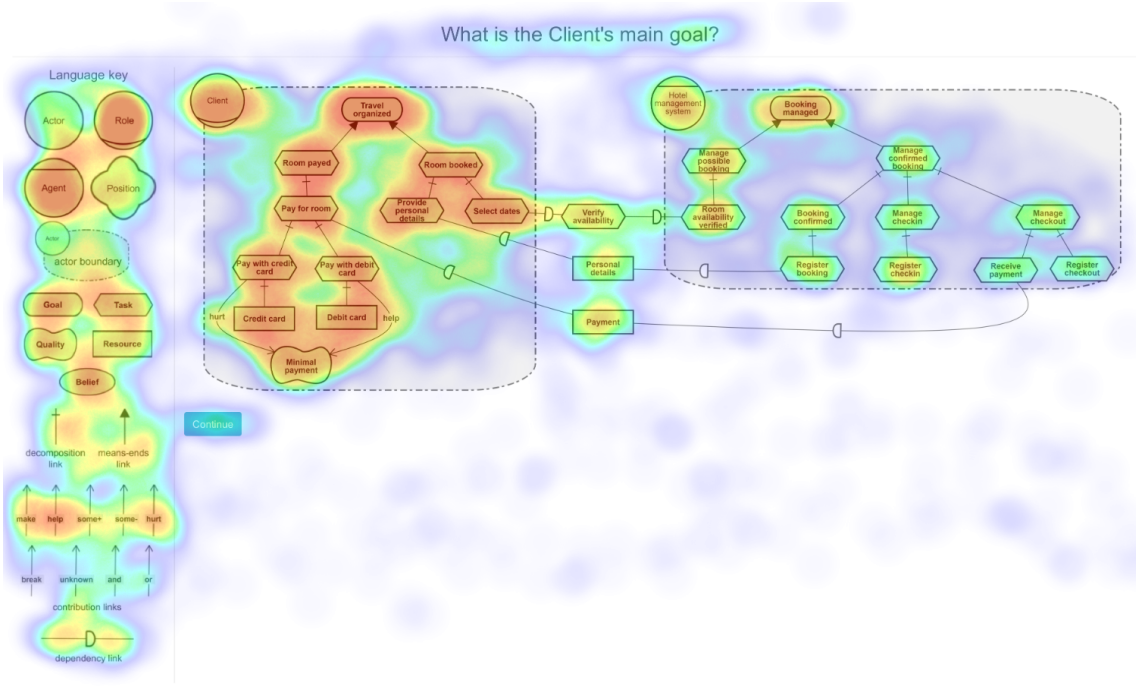
RQN3: Does a difference in the i* versions (i* 1.0 and iStar 2.0) influence the ability to* understand *i* SR models?

Assessing accuracy. Although the mean for the *precision*, *recall* and *f-measure* of participants using *i** 1.0 was lower than the ones using iStar 2.0, the differences in the distributions are not statistically significant. Nevertheless, the results for these metrics were higher than 50% in both versions. Our interpretation is that participants were able to fairly understand the models, independently on the version used. When analysing the 7 (seven) questions asked to the participants about a given *i** model, we note that the question *Which tasks are involved in checking out?* had a significantly lower number of correct answers, in both versions. This question is directly related with the *Agent Hotel management system*, and not with the *Client*. The participants focused their attention on the latter, belittling the importance of the former. This can be observed when analysing the eye-tracking data. We argue that the *Client* is perceived as the main actor, since it is the role that the participants are more familiar with. However, further studies are needed to better understand this difference.

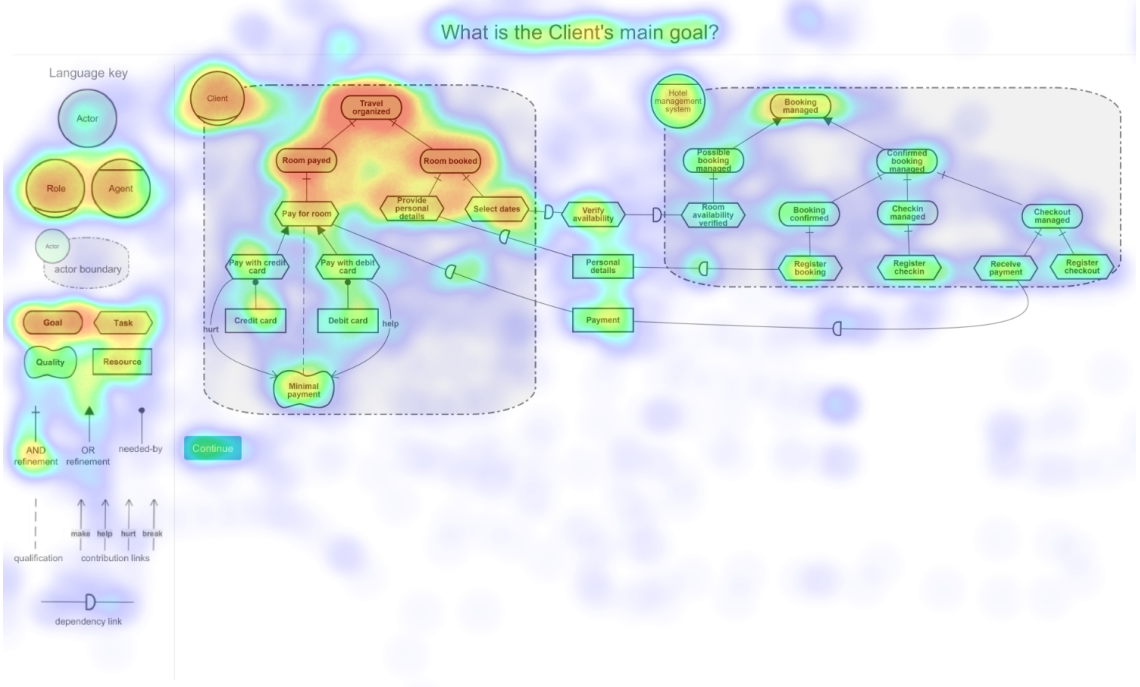
Assessing speed. There was a statistically significant difference between the versions in terms of *duration*. Participants using *i** 1.0 were ≈ 7 minutes slower to complete the task than the ones using iStar 2.0. The time for performing the *last detection* was also higher for participants using *i** 1.0 than the ones using iStar 2.0. We argue that the understanding of an *i** 1.0 model takes longer mainly due to the analysis of a higher number of model elements available in the language key. The models being analysed were similar in terms of number of elements and relationships, in both versions. However, the language key for *i** 1.0 has a higher number of model elements than the one of iStar 2.0. As it can be observed from analysing the eye-tracking data, participants using *i** 1.0 spent more time examining the language key, which may have caused the differences in terms of *speed*.

Assessing visual ease. There was a statistically significant difference between the versions in terms of visual ease. Participants using *i** 1.0 had a greater visual effort then the ones using iStar 2.0, observable through a higher *fixation rate on relevant elements*, *average duration of relevant fixations*, *average duration of irrelevant fixations*, *total number of saccades*, and *total number of saccades to the key*. Our interpretation is that, although participants looked at the right element, they also spent some time analysing all the other elements available. We argue that the higher number of model elements available in the *i** 1.0 language key may have caused this more comprehensive analysis. In Figure 6.11 we illustrate the heat maps representing the areas more frequently gazed during the

understanding task, with *i** 1.0, in Figure 6.11a; and iStar 2.0, in Figure 6.11b. The heat maps further reinforce the conclusion that participants using *i** 1.0 had a greater visual effort than the ones using iStar 2.0.



(a) Heat map for *i** 1.0 understanding task.



(b) Heat map for iStar 2.0 understanding task.

Figure 6.11: Heat maps for fixations during *i** understanding task.

Assessing mental ease. Although the mean for the *average mental workload* of participants using *i** 1.0 was slightly higher than in the ones using iStar 2.0, the difference in the distributions is not statistically significant. Nevertheless, the *average mental workload* was higher than 50% in both versions. Participants were somewhat mentally engaged on the task they were performing, independently of the version. There were some significant differences between the versions. The *average attention* was higher in participants using *i** 1.0 than in the ones using iStar 2.0, indicating a greater attention and effort while performing the task. Participants using *i** 1.0 also have a higher *average familiarity* than the ones using iStar 2.0. Some participants said to have learnt *i** 1.0 in the context of a course and working with it during a University semester, hence the initial *familiarity* with the model elements. However, this greater familiarity had no impact on the *accuracy* of the task, nor on the *mental effort* while performing it.

Assessing emotional ease. Although the mean for the *average skin conductive level* and *heart rate variability*, for both RMSSD and NN50, of participants using *i** 1.0 was slightly lower than the ones using iStar 2.0, the differences in the distributions are not statistically significant. We found no evidence on the impact of the version on the emotional ease of participants performing the understanding task on *i** models.

Assessing perceived effort. There was no statistically significant difference between the versions, in any of the NASA-TLX components. We found no evidence on the impact of the version on the perceived effort of participants performing the understanding task on *i** models.

RQN4: Does a difference in the i* versions (i* 1.0 and iStar 2.0) influence the ability to* review *i* SR models?

Assessing accuracy. Although the mean for the *precision* and *recall* of participants using *i** 1.0 was higher than the ones using iStar 2.0, the differences in the distributions are not statistically significant. Nevertheless, the *precision* and *recall* achieved by the participants was poor, being lower than 40% in both versions. The participants really struggled when reviewing *i** models, independently on the version. However, this was somewhat expected. Reviewing a model can be hard, since it involves not only reasoning about what the model represents, but also about what it does not represent (and should), and what is misrepresented. There was a statistically significant difference between the versions in terms of *f-measure*. This is due to the fact that both *precision* and *recall* where higher when using *i** 1.0. Yet, we cannot affirm that the *i** 1.0 version lead to a higher accuracy, given the results for *precision* and *recall*.

Assessing speed. There was a statistically signification difference between the versions in terms of *duration*. Participants using *i** 1.0 were ≈ 11 minutes faster to complete the task than the ones using iStar 2.0. The time for performing the *first detection* was also lower for participants using *i** 1.0 than the ones using iStar 2.0. The higher number of model elements in the *i** 1.0 language key did not hinder the speed on the review task.

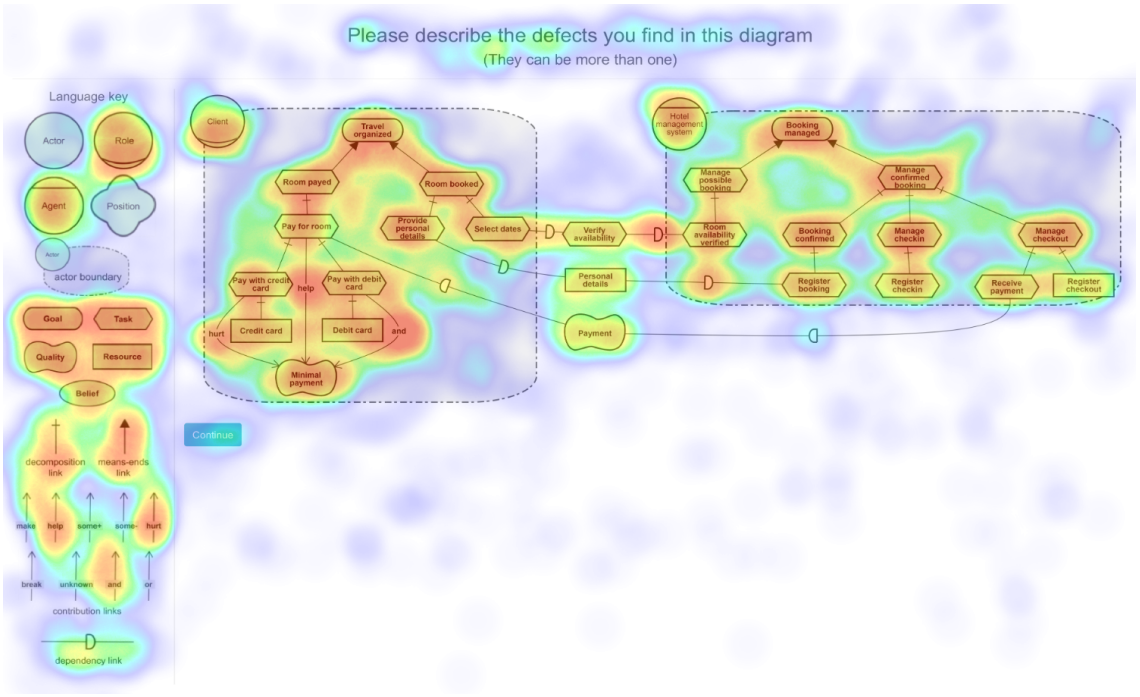
However, participants using *i** 1.0 still took ≈ 8 minutes to start answering the question. This confirms the difficulties participants encountered when reviewing *i** models.

Assessing visual ease. There was a statistically significant difference between the versions in terms of visual ease. Participants using *i** 1.0 had a greater visual effort than the ones using iStar 2.0, observable through a higher *fixation rate on relevant elements*, *fixation rate on irrelevant elements*, *total number of saccades*, and *total number of saccades to the key*. However, the *average duration of relevant fixations* and the *average duration of irrelevant fixation* were lower. Our interpretation is that, although participants were looking at the right element, they were also looking at the other elements and having difficulties deciding which ones were relevant. The models being analysed were similar in terms of number of elements and relationships, in both versions. However, the language key for *i** 1.0 has a higher number of model elements than the one of iStar 2.0. We argue that the higher number of model element available in the *i** 1.0 language key may have caused this difference, and the participants might have felt somewhat "lost" with the amount of elements to analyse. In Figure 6.12 we illustrate the heat maps representing the areas more frequently gazed during the review task, with *i** 1.0, in Figure 6.12a; and iStar 2.0, in Figure 6.12b. The heat maps further reinforce the conclusion that participants using *i** 1.0 had a greater visual effort than the ones using iStar 2.0.

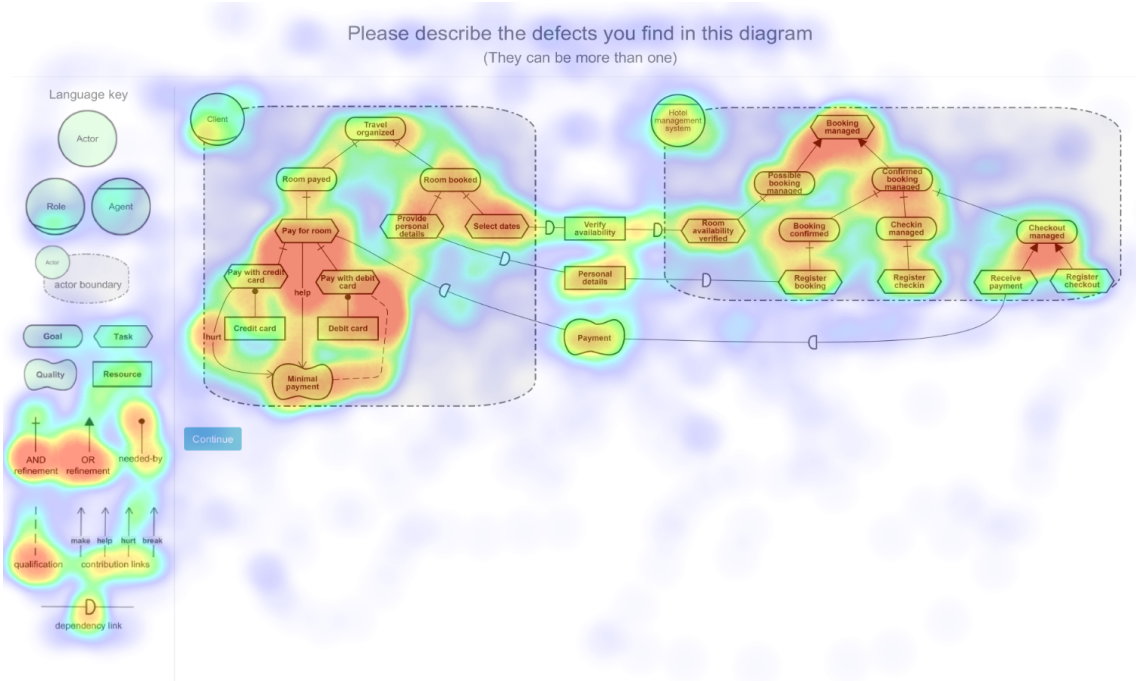
Assessing mental ease. Although the mean for the *average attention* of participants using i* 1.0 was slightly higher than in the ones using iStar 2.0, the difference in the distributions it not statistically significant. Nevertheless, the *average attention* was higher than 75% in both versions. Participants were attentive to the task they were performing. There was one significant differences between the versions. The *average mental workload* was higher in participants using *i** 1.0 than in the ones using iStar 2.0, indicating a greater effort while performing the task. We argue that this difference may be related with the higher number of elements available in *i** 1.0.

Assessing emotional ease. There was a statistically significant difference between the versions in terms of emotional ease. Participants using *i** 1.0 had a lower emotional effort than the ones using iStar 2.0, observable through a lower *average skin conductive level* and *heart rate variability* (for NN50). We argue that, although their *visual effort* was higher, were more relaxed and less stressed while performing the task. Our interpretation is that the participants did not felt the pressure for achieving a high performance, nor an evaluation apprehension.

Assessing perceived effort. Although the mean for the perceived *mental demand* of participants using *i** 1.0 was slightly lower than in the ones using iStar 2.0, the difference in the distribution is not statistically significant. Nevertheless, the perceived *mental demand* was higher than 80 (out of 100) for both versions. The participants perceived the task of modifying an *i** model, independently of the version, as being mentally challenging, which is in line with the results obtained through the biometric data. There were significant differences between the versions. The perceived *performance* of participants using *i** 1.0 was lower than the one of participants using iStar 2.0. This is congruent with



(a) Heat map for *i** 1.0 review task.



(b) Heat map for iStar 2.0 review task.

Figure 6.12: Heat maps for fixations during *i** review task.

the results obtained in terms of *accuracy*, meaning the participants were well aware of their *performance* on the task. The *frustration* of participant using *i** 1.0 was lower than of the ones using iStar 2.0. This is in line with the results obtained for emotional ease.

RQGM1: Does a difference in the level (Abby and Tim) of each GenderMag facet influence the ability to* create *i* SR models?

Assessing accuracy. There was a statistically significant difference between Abby and Tim in some of the facets, in terms of accuracy. Participants identified as Abby in the *information processing* and *risk* facets had a higher *precision*, when compared with those identified as Tim. However, *recall* for Abby in the *risk* facet was lower. Our interpretation is that Tim is able to achieve a higher recall because he is risk-tolerant, and takes a chance even when he is not sure. Yet, this causes his *precision* to be lower. Abby is risk-averse and only answers when she's sure. As such, when she answers, her answer tends to be correct, but incomplete (she does not add a model element if she is not absolutely confident). As such, the *complexity* of the *i** SR models created by Abby in the *risk* facet was lower, but the *completeness* of those models was lower as well.

Assessing speed. There was a statistically significant difference between Abby and Tim in almost all of the facets, in terms of speed. Participants identified as Abby in the *learning style* facet were slower than the ones characterised as Tim, taking ≈ 10 minutes more in the overall *duration* of the task. However, Abby in the *self-efficacy* facet took less time to complete the task. Our interpretation for the latter is that, without someone to first show how tasks of this type could be performed, Abby felt she had already given her best and decided to finish the task earlier. For the former, since Tim tends to have a tinkering approach, this may help him to be faster. Tim in the *risk* and *learning style* facets makes the *first action* in the model really early. In fact, he starts trying to solve the task even before finishing reading the problem description. As for Abby, she only starts after some time. Finally, in the *self-efficacy* facet, the *processing duration* was lower for Tim. This means that, after the creation of the model, Tim submits it without performing a revision. We argue that this is due to his high confidence on his work.

Assessing visual ease. There was a statistically significant difference between Abby and Tim in some of the facets, in terms of visual ease. Participants characterised as Abby in the *information processing* and *self-efficacy* facets had a greater visual effort, observable through a higher *fixation rate on irrelevant elements* and *average duration of irrelevant fixations*. However, Abby had a lower *average duration of relevant fixations*. Our interpretation is that Tim, being more selective in the way he processes information, is able to focus more on the relevant elements. As for Abby, she tends to further analyse the information provided, hence the focus on the irrelevant elements. On the other hand, Tim on the *learning style* facet had a higher *fixation rate on irrelevant* elements. We argue that, since Tim tends to have a tinkering approach, he's experimenting and focusing on several elements, even if they are irrelevant.

Assessing mental ease. There was a statistically significant difference between Abby and Tim in some of the facets, in terms of mental ease. Participants characterised as Abby in the *information processing* and *self-efficacy* facets had a greater mental effort, observable through a higher *average attention* and *average mental workload*. Since Abby is more comprehensive when processing information, her level of attention indicates she is highly engaged in the task. Similarly, given that she as a low *self-efficacy*, her mental workload becomes higher, indicating effort while performing the task.

Assessing emotional ease. There was a statistically significant difference between Abby and Tim in the *risk* facet, in terms of emotional ease. Participants characterised as Tim had a higher *heart rate variability*, for *RMSSD*, than Abby. Since Tim is risk-tolerant, our interpretation is that Tim was excited when performing the task.

Assessing perceived effort. There was a statistically significant difference between Abby and Tim in all the facets, in terms of perceived effort. Participants characterised as Abby in the *learning style* and *risk* facets had a higher perceived *physical demand* than the ones characterised as Tim. The perceived *temporal demand* was also higher for Abby in the *self-efficacy*, *learning style*, and *risk* facets. The perceived *mental demand* was higher for Abby in the *information processing* facet. Finally, the *frustration* was higher for Abby in all the facets. This is in line with the results obtained in terms of *accuracy*, *speed*, and biometric data, meaning the participants were well aware of their *performance* and *effort* on the task.

RQGM2: Does a difference in the level (Abby and Tim) of each GenderMag facet influence the ability to* modify *i* SR models?

Assessing accuracy. There was a statistically significant difference between Abby and Tim in some of the facets, in terms of accuracy. Participants identified as Abby in the *motivation*, *self-efficacy*, *information processing*, and *risk* facets had a higher *precision*, when compared with those identified as Tim. However, there were no differences in terms of *recall*. Still, some patterns emerged, as in the creation task. Abby is risk-averse and only answers when she's sure. Her answer tends to be correct, but incomplete (she does not change, add, or remove a model element if she is not absolutely confident). As such, the *complexity* of the *i** SR models modified by Abby in the *risk* facet was lower, but the *completeness* of those models was lower as well. In fact, in the *risk* facet, Abby tended to make fewer changes than Tim.

Assessing speed. There was a statistically significant difference between Abby and Tim in almost all the facets, in terms of speed. Participants identified as Abby in the *learning style* facet were slower than the ones characterised as Tim, taking ≈ 8 minutes more in the overall *duration* of the task. However, Abby in the *self-efficacy* facet took less time to complete the task. Our interpretation for the latter is that, without someone to first show how tasks of this type could be performed, Abby felt she had already given her best and decided to finish the task earlier. For the former, since Tim tends to have

a tinkering approach, this may help him to be faster. Tim in the *risk* and *learning style* facets makes the *first action* in the model really early. In fact, he starts trying to solve the task even before finishing reading the problem description. As for Abby, she only starts after some time. Finally, in the *self-efficacy* facet, the *processing duration* was lower for Tim. This means that, after the modification of the model, Tim submits it without performing a revision. We argue that this is due to his high confidence on his work.

Assessing visual ease. There was a statistically significant difference between Abby and Tim in almost all the facets, in terms of visual ease. Participants characterised as Abby in the *information processing*, *self-efficacy*, and *learning style* facets had a greater visual effort, observable through a higher *fixation rate on irrelevant elements* and *average duration of irrelevant fixations*. However, Abby had a lower *average duration of relevant fixations*. Our interpretation is that Tim, being more selective in the way he processes information, is able to focus more on the relevant elements. As for Abby, she tends to further analyse the information provided, hence the focus on the irrelevant elements. Furthermore, participants characterised as Abby in the *information processing* and *risk* facets had a higher *total number of saccades to the key*. We argue that Abby looked more to the key in order to make sure she was selecting the right model element.

Assessing mental ease. There was a statistically significant difference between Abby and Tim in some of the facets, in terms of mental ease. Participants characterised as Abby in the *information processing* facet had a greater mental effort, observable through a higher *average attention* and *average mental workload*. Since Abby is more comprehensive when processing information, her level of attention indicates she is highly engaged in the task.

Assessing emotional ease. There was a statistically significant difference between Abby and Tim in the *risk* facet, in terms of emotional ease. Participants characterised as Tim had a higher *heart rate variability*, for *RMSSD*, than the ones identified as Abby. Since Tim is risk-tolerant, our interpretation is that Tim was excited when performing the task. On the other hand, participants identified as Abby in the *information processing* style had a higher *heart rate variability*, for *RMSSD*, than the ones characterised as Tim. Given that Abby is more comprehensive when processing information, we argue that the amount of model elements to analyse and possibly change might have made her to feel more stressed and anxious.

Assessing perceived effort. There was a statistically significant difference between Abby and Tim some of the facets, in terms of perceived effort. Participants characterised as Abby in the *information processing*, *self-efficacy* and *risk* facets had a higher perceived *physical demand* than the ones characterised as Tim. The perceived *temporal demand* was also higher for Abby in the *risk* facet. The perceived *mental demand* was higher for Abby in the *information processing* and *self-efficacy* facets. Finally, the *frustration* was higher for Abby in all the facets. This is in line with the results obtained in terms of *speed*, and biometric data, meaning the participants were well aware of their effort on the task.

RQGM3: Does a difference in the level (Abby and Tim) of each GenderMag facet influence the ability to* understand *i* SR models?

Assessing accuracy. There was a statistically significant difference between Abby and Tim in some of the facets, in terms of accuracy. Participants characterised as Abby in the *information processing* and *risk* facets has a higher *precision*, when compared with those identified as Tim. Furthermore, Abby in the *information processing* facet also had a higher *recall*. There were no differences in terms of *recall* for the risk facet. Our interpretation is that analysing the *i** SR model comprehensively helped Abby to better understand it.

Assessing speed. There was a statistically significant difference between Abby and Tim in all of the facets, in terms of speed. Participants characterised as Abby in the *learning style* and *motivation* facets were slower than the ones characterised as Tim, taking ≈ 4 minutes more in the overall *duration* of the task. Since Tim tends to have a tinkering approach, this may help him to be faster. Tim in the *risk*, *learning style* and *self-efficacy* facets makes the *first detection* in the model really early. In fact, he starts trying to solve the task even before finishing reading the problem description. As for Abby, she only starts after some time. Finally, in the *information processing* facet, the *processing duration* is higher for Abby. Our interpretation is that, since Abby is comprehensive when analysing information, she prefers to revise the model to make sure that nothing was forgotten.

Assessing visual ease. There was a statistically significant difference between Abby and Tim in almost all of the facets, in terms of visual ease. Participants characterised as Abby in the *information processing*, *self-efficacy*, and *learning style* facets had a greater visual effort, observable through a higher *fixation rate on irrelevant elements* and *average duration of irrelevant fixations*. However, Abby had a lower *average duration of relevant fixations*. Our interpretation is that Tim, being more selective in the way he processes information, is able to focus more on the relevant elements. As for Abby, she tends to further analyse the information provided, hence the focus on the irrelevant elements. Furthermore, participants characterised as Abby in the *learning style* facet had a higher *total number of saccades to the key*. We argue that Abby looked more to the key in order to make sure she completely understood the elements in the element, in order to be able to select the correct one.

Assessing mental ease. There was a statistically significant difference between Abby and Tim in some of the facets, in terms of mental ease. Participants characterised as Abby in the *information processing* and *self-efficacy* facets had a greater mental effort, observable through a higher *average attention* and *average mental workload*. Since Abby is more comprehensive when processing information, her level of attention indicates she is highly engaged in the task. Similarly, given that she as a low *self-efficacy*, her mental workload becomes higher, indicating effort while performing the task.

Assessing emotional ease. There was a statistically significant difference between Abby and Tim in the *risk* facet, in terms of emotional ease. Participants characterised as Tim had a higher *heart rate variability*, for *RMSSD*, than the ones identified as Abby. Since Tim is risk-tolerant, our interpretation is that Tim was excited when performing the task.

On the other hand, participants identified as Abby in the *information processing* style had a higher *heart rate variability*, for *RMSSD*, than the ones characterised as Tim. Given that Abby is more comprehensive when processing information, we argue that the amount of model elements to analyse might have made her to feel more stressed and anxious.

Assessing perceived effort. There was a statistically significant difference between Abby and Tim in all the facets, in terms of perceived effort. Participants characterised as Abby in the *motivation*, *self-efficacy*, *learning style*, *information processing*, and *risk* facets had a higher perceived *physical demand* than the ones characterised as Tim. However, it was lower than 40 (out of 100) for both Abby and Tim. The perceived *temporal demand* was also higher for Abby in the *motivation*, and *risk* facets. The perceived *mental demand* was higher for Abby in the *motivation*, *self-efficacy*, and *information processing* facets. Finally, the *frustration* was higher for Abby in the *motivation* and *risk* facets, while the perceived *effort* was higher for Abby in the *information processing* facet. This is in line with the results obtained in terms of *speed*, and biometric data, meaning the participants were well aware of their effort on the task.

RQGM4: Does a difference in the level (Abby and Tim) of each GenderMag facet influence the ability to* review *i* SR models?

We found no evidence that the *learning style* facet influences the accuracy, speed, ease (visual, mental, and emotional), and perceived effort when performing the review task on *i** SR models.

Assessing accuracy. There was a statistically significant difference between Abby and Tim in some of the facets, in terms of accuracy. Participants identified as Abby in the *information processing*, *self-efficacy*, and *risk* facets has a higher *precision*, when compared with those identified as Tim. However, *recall* for Abby in the *risk* facet was lower. Our interpretation is that Tim is able to achieve a higher recall because he is risk-tolerant, and takes a chance even when he is not sure. Yet, this causes his *precision* to be lower. Abby is risk-averse and only answers when she's sure. As such, when she answers, her answer tends to be correct, but incomplete.

Assessing speed. There was a statistically significant difference between Abby and Tim in some of the facets, in terms of speed. Participants identified as Abby in the *self-efficacy* facet took less time to complete the task than Tim. Our interpretation is that, without someone to first show how tasks of this type could be performed, Abby felt she had already given her best and decided to finish the task earlier. Tim in the *risk* and *learning style* facets makes the *first action* in the model really early. In fact, he starts trying to solve the task even before finishing reading the problem description. As for Abby, she only starts after some time.

Assessing visual ease. There was a statistically significant difference between Abby and Tim in some of the facets, in terms of visual ease. There was a statistically significant difference between Abby and Tim in almost all of the facets, in terms of visual ease.

Participants characterised as Abby in the *information processing*, *self-efficacy*, and *risk* facets had a greater visual effort, observable through a higher *fixation rate on irrelevant elements* and *average duration of irrelevant fixations*. However, Abby had a lower *average duration of relevant fixations*. Our interpretation is that Tim, being more selective in the way he processes information, is able to focus more on the relevant elements. As for Abby, she tends to further analyse the information provided, hence the focus on the irrelevant elements.

Assessing mental ease. There was a statistically significant difference between Abby and Tim in some of the facets, in terms of mental ease. Participants characterised as Abby in the *information processing* and *self-efficacy* facets had a greater mental effort, observable through a higher *average attention* and *average mental workload*. Since Abby is more comprehensive when processing information, her level of attention indicates she is highly engaged in the task.

Assessing emotional ease. There was a statistically significant difference between Abby and Tim in some of the facets, in terms of emotional ease. Participants identified as Abby in the *information processing* style had a higher *heart rate variability*, for *NN50*, than the ones characterised as Tim. Given that Abby is more comprehensive when processing information, we argue that the amount of model elements to analyse might have made her to feel more stressed and anxious.

Assessing perceived effort. There was a statistically significant difference between Abby and Tim in some of the facets, in terms of perceived effort. Participants characterised as Abby in the *self-efficacy* and *risk* facets had a higher perceived *temporal demand* than the ones characterised as Tim. The perceived *mental demand* was higher for Abby in the *self-efficacy* facet. Finally, the *frustration* was higher for Abby in the *self-efficacy* and *risk* facets, while the perceived *effort* was higher for Abby in the *self-efficacy* facet. This is in line with the results obtained in terms of biometric data, meaning the participants were well aware of their effort on the task.

6.4.2 Inferences

It's not easy to review an *i SR model.** Our participants really struggled when reviewing *i** models, independently on the version. However, this was somewhat expected. Reviewing a model can be hard, since it involves not only reasoning about what the model represents, but also about what it does not represent (and should), and what is misrepresented. In general, participants had little to no prior knowledge on the *i** version, although some participants had learnt *i** 1.0 in the context of a course. With training, we are confident that participants would be able to achieve a higher performance. However, the obtained results can also mean that *i** is possibly not a good suit for communication with stakeholders not knowledgeable on the version, even through the results for the understanding task were good.

The iStar 2.0 version outperformed *i 1.0.** For the majority of the metrics, participants were able to achieve a better performance and lower effort when using iStar 2.0 than the participants using *i** 1.0. We argue that the lower number of actors and contribution links helped novice participants to better understand iStar 2.0. Further studies are needed to analyse if these results hold when the participants are experts on the versions.

Information processing and risk have impact on accuracy. Participants identified as Abby in these facets were able to achieve an acceptable level of precision, even without much training. However, her attitude towards risk is undermining the recall. We argue that, with training, Abby would become more confident in her skills and could achieve great results for both precision and recall. As for Tim, making him aware that risking too much is possibly sabotaging his results could help with his precision.

Risk has impact on speed. Participants characterised as Tim in this facet tend to be faster, start trying to solve the task even before finishing reading the problem description. Moreover, he submits the answer without any further revision. We argue that a lower duration is not always a desirable outcome, if it compromises the accuracy of task, which we interpret has being the Tim's case. In particular, by not revising the model, Tim may be losing an opportunity for improvement of his answer and for a higher precision.

Information processing, self-efficacy and risk have impact on ease. Abby in these facets has a more comprehensive analysis of the problem description and the model elements available in the editor's toolbar or in the language key. The visual effort, attention and mental workload is higher due to this thorough inspection. Plus, in general, Abby is more engaged at the task she's performing. Tim, however, is able to better separate what is relevant from what is not. We argue that, in this particular scenario, having a higher effort is not perceived as being harmful. Nonetheless, being able to more precisely understand what is relevant is a great advantage in terms of effort.

The iStar 2.0 version is better suited for Abby. Participants characterised as Abby, independently on the facet, were able to have a better performance and lower effort when using iStar 2.0. In particular for the information processing facet, we argue that the lower number of actors and contributions links in the iStar 2.0 version was an important factor for Abby, especially in terms of mental workload. For Tim, on the other hand, there was not significant difference between the *i** versions.

People diversity is key. When analysing the GenderMag results, we note that complementarity of the results achieved by Tim and Abby suggests that, rather than targeting the requirements process to one of them, there is more to be gained in leveraging their diversity. One possible way of doing so would be to build up teams with this diversity, specially in terms of *information processing*, *self-efficacy* and *risk*.

6.5 Summary

We performed a family of quasi-experiments to analyse the impact of different *i** versions, as well as different levels in each of the five GenderMag facets, when creating, modifying,

understanding and reviewing *i** SR models. We measured the accuracy, speed, ease (visual, mental, and emotional), and perceived effort of a total of 340 participants. We used metrics of task success, time, and effort, collected with eye-tracking, EEG and EDA sensors, and participants' feedback through a NASA-TLX questionnaire.

We found that reviewing *i** SR model was a challenge for our participants, independently on the *i** version. However, our participants were able to achieve better results with iStar 2.0 than with *i** 1.0. Furthermore, there are several differences in the individual characteristics (the GenderMag facets) of the participants that had an influence on their performance and effort.

Chapter 7

Evaluation of ARNE and ALCO Use Cases

In this Chapter, we start by presenting the experimental protocol used in the ARNE and ALCO use cases quasi-experiments, following Jedlitschka *et al.* guidelines [113] on how to report (quasi-)experiments in Software Engineering. The Chapter provides further details on the general experimental protocol previously presented in Chapter 5. It covers planning, execution, analysis, and discussion on the results and their implications, thus presenting a complete description of the empirical research performed for evaluating the appropriateness recognisability and learnability of use cases templates. Although some Subsections are common to all the quasi-experiments, and fully described in Chapter 5, we decided to maintain the placeholders here, and make the reference to the corresponding Subsection on Chapter 5, for organisation purposes.

7.1 Experiments Planning

7.1.1 Goals

We describe our research goals using the GQM research goal template [7, 8]. We analyse differences in 2 (two) main sets, related with: use case templates, and levels of the GenderMag facets. Each set has 4 (four) main goals, each related with the tasks performed by the participants: creation, modification, understanding, and reviewing. Finally, each high level goal has a set of sub-goals, related with accuracy, speed, and ease, which are also defined. All the goals are similar, only changing the *underline and italic* part. However, they are fully specified for documentation purposes and easier reference.

The first set of goals is related with the **use case templates** (**GT**) themselves. The objective is to compare the differences between the results achieved when using ARNE and ALCO templates.

(GT1) **Analyse** differences in the use case templates, **for the purpose of** evaluation, **with respect to** their effects on the *creation* of use case specifications, **from the viewpoint of** researchers, **in the context of** experiments conducted at our University and at software companies.

(GT1.1) **Analyse** differences in the use case templates, **for the purpose of** evaluation, **with respect to** their effects on the *accuracy to create* use case specifications, **from the viewpoint of** researchers, **in the context of** experiments conducted at our University and at software companies.

(GT1.2) **Analyse** differences in the use case templates, **for the purpose of** evaluation, **with respect to** their effects on the *speed to create* use case specifications, **from the viewpoint of** researchers, **in the context of** experiments conducted at our University and at software companies.

(GT1.3) **Analyse** differences in the use case templates, **for the purpose of** evaluation, **with respect to** their effects on the *ease to create* use case specifications, **from the viewpoint of** researchers, **in the context of** experiments conducted at our University and at software companies.

(GT2) **Analyse** differences in the use case templates, **for the purpose of** evaluation, **with respect to** their effects on the *modification* of use case specifications, **from the viewpoint of** researchers, **in the context of** experiments conducted at our University and at software companies.

(GT2.1) **Analyse** differences in the use case templates, **for the purpose of** evaluation, **with respect to** their effects on the *accuracy to modify* use case specifications, **from the viewpoint of** researchers, **in the context of** experiments conducted at our University and at software companies.

(GT2.2) **Analyse** differences in the use case templates, **for the purpose of** evaluation, **with respect to** their effects on the *speed to modify* use case specifications, **from the viewpoint of** researchers, **in the context of** experiments conducted at our University and at software companies.

(GT2.3) **Analyse** differences in the use case templates, **for the purpose of** evaluation, **with respect to** their effects on the *ease to modify* use case specifications, **from the viewpoint of** researchers, **in the context of** experiments conducted at our University and at software companies.

(GT3) **Analyse** differences in the use case templates, **for the purpose of** evaluation, **with respect to** their effects on the *understanding* of use case specifications, **from the viewpoint of** researchers, **in the context of** experiments conducted at our University and at software companies.

(GT3.1) **Analyse** differences in the use case templates, **for the purpose of** evaluation, **with respect to** their effects on the *accuracy to understand* use case specifications, **from the viewpoint of** researchers, **in the context of** experiments conducted at our University and at software companies.

(GT3.2) **Analyse** differences in the use case templates, **for the purpose of** evaluation, **with respect to** their effects on the *speed to understand* use case specifications, **from the viewpoint of** researchers, **in the context of** experiments conducted at our University and at software companies.

(GT3.3) **Analyse** differences in the use case templates, **for the purpose of** evaluation, **with respect to** their effects on the *ease to understand* use case specifications, **from the viewpoint of** researchers, **in the context of** experiments conducted at our University and at software companies.

(GT4) **Analyse** differences in the use case templates, **for the purpose of** evaluation, **with respect to** their effects on the *reviewing* of use case specifications, **from the viewpoint of** researchers, **in the context of** experiments conducted at our University and at software companies.

(GT4.1) **Analyse** differences in the use case templates, **for the purpose of** evaluation, **with respect to** their effects on the *accuracy to review* use case specifications, **from the viewpoint of** researchers, **in the context of** experiments conducted at our University and at software companies.

(GT4.2) **Analyse** differences in the use case templates, **for the purpose of** evaluation, **with respect to** their effects on the *speed to review* use case specifications, **from the viewpoint of** researchers, **in the context of** experiments conducted at our University and at software companies.

(GT4.3) **Analyse** differences in the use case templates, **for the purpose of** evaluation, **with respect to** their effects on the *ease to review* use case specifications, **from the viewpoint of** researchers, **in the context of** experiments conducted at our University and at software companies.

The second set of goals is related with the **levels of the GenderMag facets** (GGM). The objective is to compare the differences between the personas (Abby and Tim) on each of the 5 (five) facets (motivation for using software, information processing style, computer self-efficacy, attitude towards risk, and ways of learning new technology).

(GGM1) **Analyse** differences in the level of the GenderMag facets, **for the purpose of** evaluation, **with respect to** their effects on the *creation* of use case specifications, **from the viewpoint of** researchers, **in the context of** experiments conducted at our University and at software companies.

(GGM1.1) **Analyse** differences in the level of the GenderMag facets, **for the purpose of** evaluation, **with respect to** their effects on the *accuracy to create* use case specifications, **from the viewpoint of** researchers, **in the context of** experiments conducted at our University and at software companies.

(GGM1.2) **Analyse** differences in the level of the GenderMag facets, **for the purpose of** evaluation, **with respect to** their effects on the *speed to create* use case specifications, **from the viewpoint of** researchers, **in the context of** experiments conducted at our University and at software companies.

(GGM1.3) **Analyse** differences in the level of the GenderMag facets, **for the purpose of** evaluation, **with respect to** their effects on the *ease to create* use case specifications, **from the viewpoint of** researchers, **in the context of** experiments conducted at our University and at software companies.

(GGM2) **Analyse** differences in the level of the GenderMag facets, **for the purpose of** evaluation, **with respect to** their effects on the *modification* of use case specifications, **from the viewpoint of** researchers, **in the context of** experiments conducted at our University and at software companies.

(GGM2.1) **Analyse** differences in the level of the GenderMag facets, **for the purpose of** evaluation, **with respect to** their effects on the *accuracy to modify* use case specifications, **from the viewpoint of** researchers, **in the context of** experiments conducted at our University and at software companies.

(GGM2.2) **Analyse** differences in the level of the GenderMag facets, **for the purpose of** evaluation, **with respect to** their effects on the *speed to modify* use case specifications, **from the viewpoint of** researchers, **in the context of** experiments conducted at our University and at software companies.

(GGM2.3) **Analyse** differences in the level of the GenderMag facets, **for the purpose of** evaluation, **with respect to** their effects on the *ease to modify* use case specifications, **from the viewpoint of** researchers, **in the context of** experiments conducted at our University and at software companies.

(GGM4) **Analyse** differences in the level of the GenderMag facets, **for the purpose of** evaluation, **with respect to** their effects on the *reviewing* of use case specifications, **from the viewpoint of** researchers, **in the context of** experiments conducted at our University and at software companies.

(GGM4.1) **Analyse** differences in the level of the GenderMag facets, **for the purpose of** evaluation, **with respect to** their effects on the *accuracy to review* use case specifications, **from the viewpoint of** researchers, **in the context of** experiments conducted at our University and at software companies.

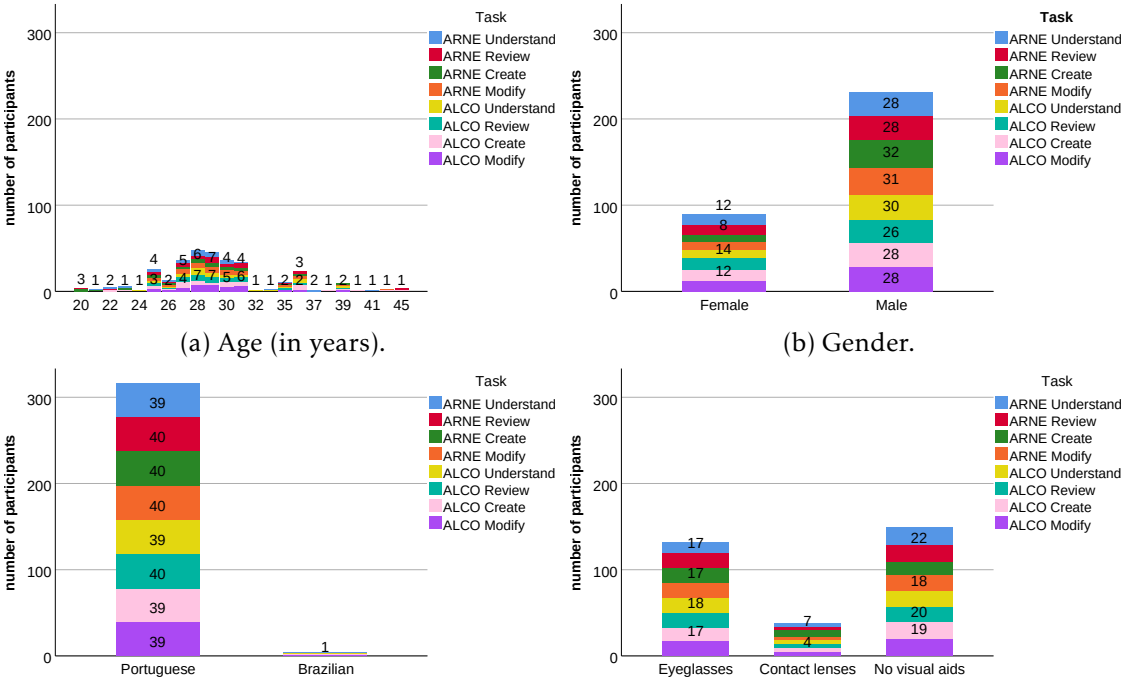
(GGM4.2) **Analyse** differences in the level of the GenderMag facets, **for the purpose of** evaluation, **with respect to** their effects on the *speed to review* use case

specifications, **from the viewpoint of** researchers, **in the context of** experiments conducted at our University and at software companies.

(GGM4.3) Analyse differences in the level of the GenderMag facets, **for the purpose of** evaluation, **with respect to** their effects on the *ease to review* use case specifications, **from the viewpoint of** researchers, **in the context of** experiments conducted at our University and at software companies.

7.1.2 Participants

We had 160 participants using the *ARNE* template, and 160 using *ALCO* template, in a total of 320 participants. Concerning participants *age* distribution (Figure 7.1a), they had between 20 and 45 years old, with an average of 29 years old. With respect to *gender* (Figure 7.1b), there were 231 male participants and 89 females. In terms of *nationality* (Figure 7.1c), 316 were Portuguese and 4 were Brazilian. Regarding the *usage of reading devices* (Figure 7.1d), 132 participants wore eyeglasses and 38 had contact lenses.



(a) Age (in years).

(b) Gender.

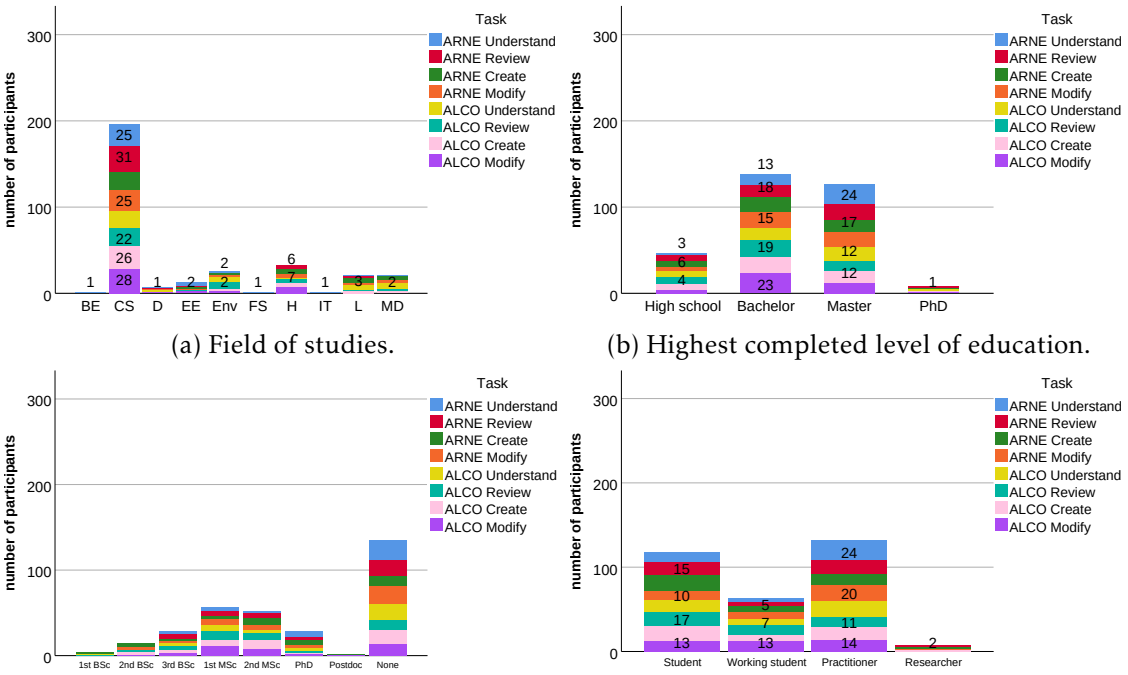
(c) Nationality.

(d) Usage of reading devices.

Figure 7.1: Participants general demographic information.

All participants had some university level training. Their *field of studies* (Figure 7.2a) spanned across multiple areas. We had 1 biomedical engineer (BE), 196 computer scientists (CS), 7 designers (D), 13 electrotechnical engineers (EE), 26 environmental engineers (Env), 1 forensic scientist (FS), 33 historians (H), 1 information technologist (IT), 21 lawyers (L), and 21 medical doctors (MD). For *highest completed level of education* (Figure 7.2b), 47 completed high school, 138 concluded a BSc, 127 had a MSc, and 8 a PhD degree. Concerning *current level of education* (Figure 7.2c), 4 were in the first year of the

BSc degree, 15 on the second year, and 28 on the third and final year. As for MSc students, 56 were in the first year, and 52 were on the second and final year. Finally, 28 were doing a PhD, 2 were doing a Post-Doc and 135 were no longer studying. The ones that were no longer studying had at least 4 years of experience. With respect to *current occupation* (Figure 7.2d), 117 of the participants were students, 63 were working students, 132 were practitioners, and 8 were researchers.



(a) Field of studies.

(b) Highest completed level of education.

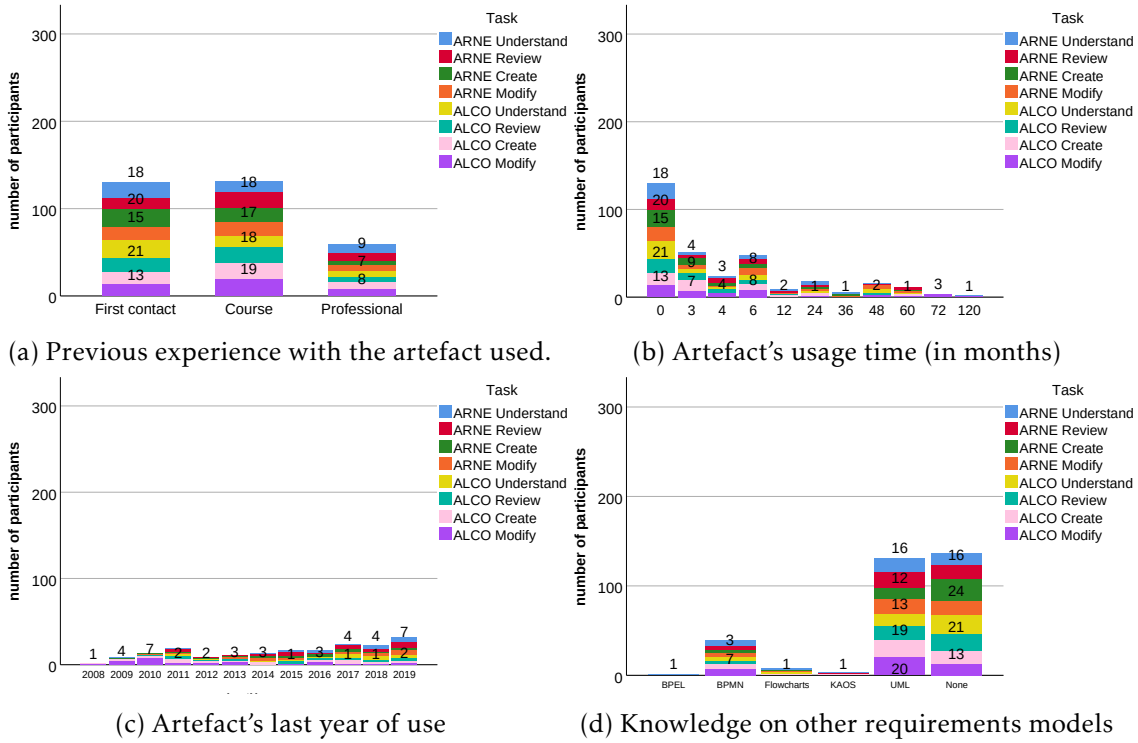
(c) Current level of education.

(d) Current occupation.

Figure 7.2: Participants academic and professional demographic information.

Regarding previous *experience* (Figure 7.3a) with the use case template used in the task, for 130 participants it was their first contact with the template. However, 131 learnt it in the context of a course, and 59 in a professional environment. In those two latter scenarios, participants *usage time* with the templates (Figure 7.3b) had an average of 10 months. Participants tend to refer to the last usage time in terms of full years (for example, saying one or two years ago, and never one and a half years ago). On the other hand, some participants referred to 3, 4 or 6 months. We argue that all those months correspond to a University semester, depending on how people count. As for the *last use* of the template (Figure 7.3c), 33 participants were still using it in their daily work when the studies were conducted. Lastly, in terms of *knowledge on other requirements models* (Figure 7.3d), 131 participants claim to know UML in general, 39 referred to BPMN, 8 specifically said to work with flowcharts in particular, 4 mentioned KAOS, and 1 BPEL. The remaining 137 participants didn't report knowing any requirements language.

Participants spanned a reasonably wide range of values of each of the GenderMag facets, with 3 participants being characterised as a "pure" Abby and 3 as a "pure" Tim



(a) Previous experience with the artefact used.

(b) Artefact's usage time (in months)

(c) Artefact's last year of use

(d) Knowledge on other requirements models

Figure 7.3: Participants knowledge on requirements models.

(Figures 7.4a and 7.4b). The other 314 participants had mixed characteristics of both Abby and Tim.

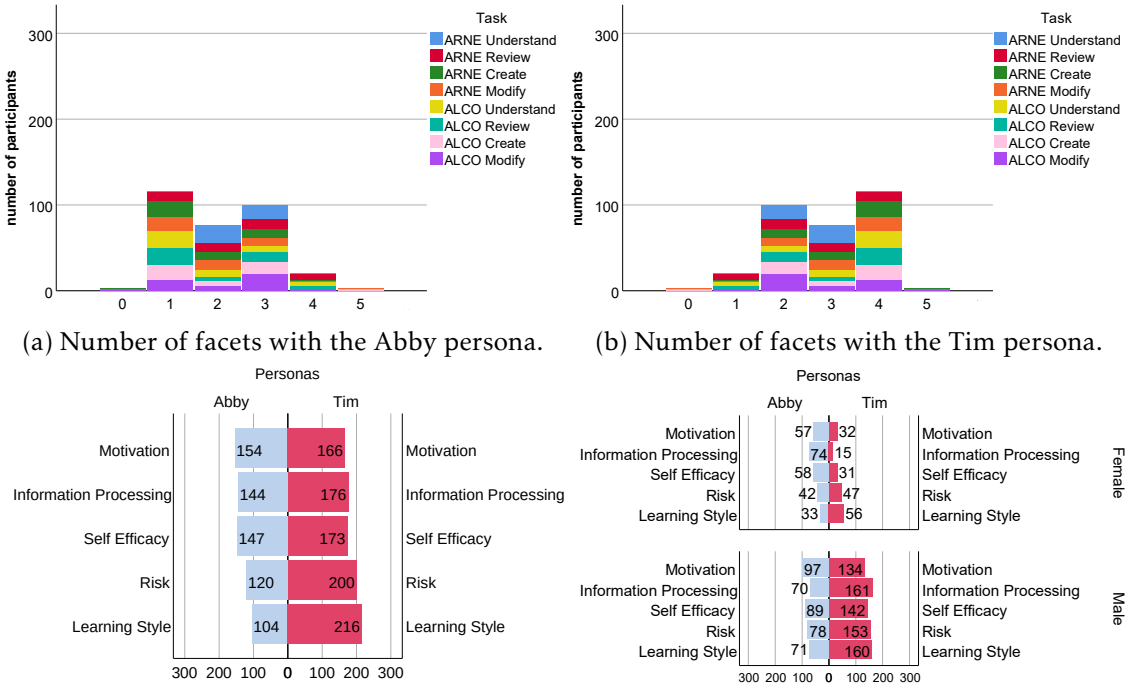
When analysing each facet (Figure 7.4c), the majority of the participants was identified as Tim in all the facets. However, there is a greater difference between the personas in the *risk* and in the *learning style* facets.

Taking a closer look into the relationship between the persona in each of the facets and the gender of participants (Figure 7.4d), the majority of female participants was characterised as Abby in all the facets, being *risk* and *learning style* an exception. As for the males, the majority of participants was classified as Tim in all the facets. These results support the literature claim [10, 191] that characteristics in how people solve problems often cluster by gender.

7.1.3 Experimental Materials

The experimental materials included *(i)* a participant consent form; *(ii)* a video of fish swimming; *(iii)* a video tutorial about the artefact; *(iv)* a problem description for the creation task; or a problem description, an initial model and a new requirement for the modification task; *(v)* a NASA-TLX questionnaire; *(vi)* a demographic questionnaire; and *(vii)* a GenderMag questionnaire. The materials *(i)*, *(ii)*, *(v)*, *(vi)* and *(vii)* were previously described in Section 5.1.3 of Chapter 5.

The **video tutorial**, with 2 minutes and 11 seconds for ARNE and 2 minutes and 40 seconds for ALCO, explained the elements of an ARNE or ALCO template, depending



(a) Number of facets with the Abby persona.

(b) Number of facets with the Tim persona.

(c) Personas in each facet.

(d) Personas in each facet, by gender.

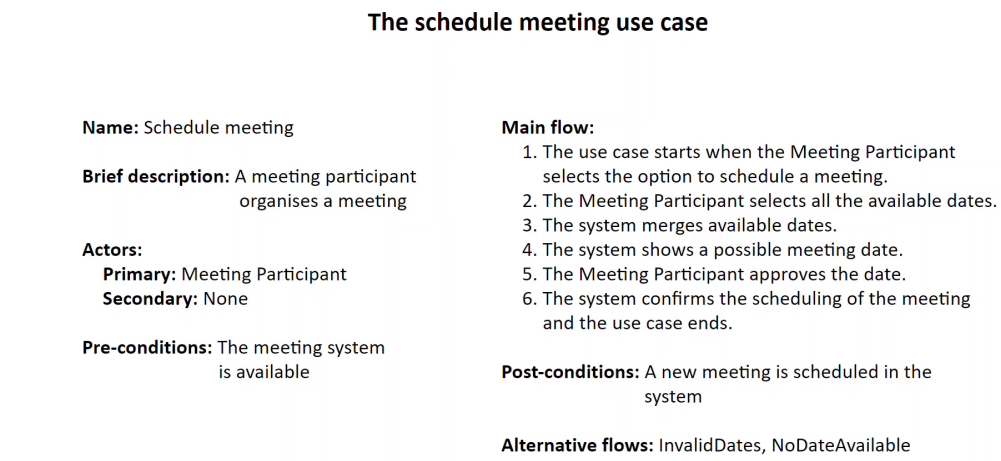
Figure 7.4: Participants distribution across GenderMag facets.

on the artefact that was evaluated on that quasi-experiment. The tutorials have different durations, proportional to the number of elements and concepts in each use case template. The tutorial included the specification of a correct use case (similar to those that were going to be created, modified, understood or revised in the quasi-experiment) about a meeting scheduler system; and an audio and textual description of both tags, as they are being introduced, and their role in the use case under construction. The participants had no control over the video, not being able to pause it or resume it, since having different viewing times and going through specific parts of the tutorial more than one time could impact the results. A snapshot is presented in Figure 7.5a, for ARNE; and in Figure 7.5b, for ALCO.

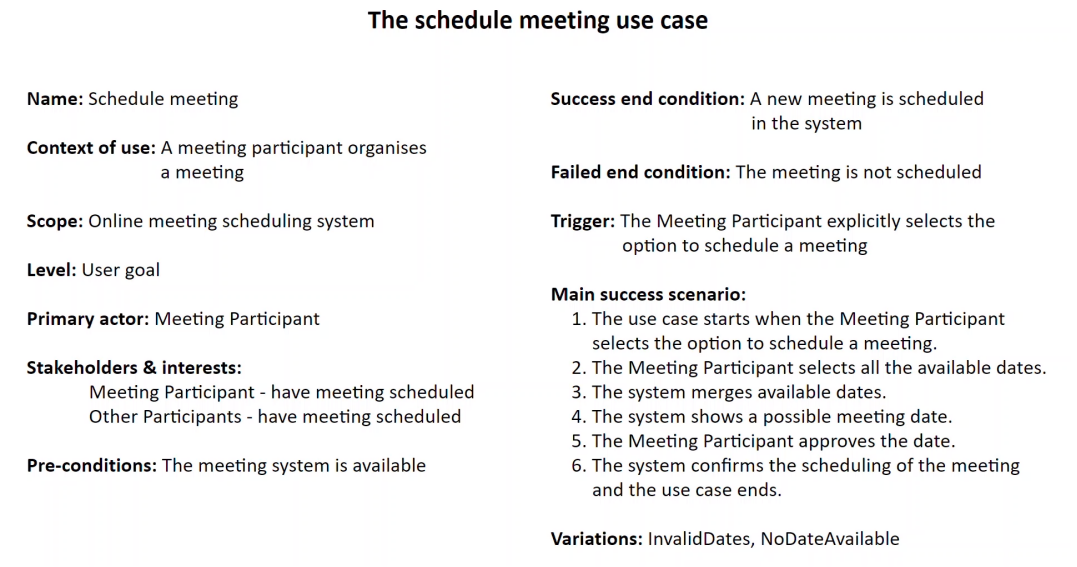
In terms of **tasks**, we prepared 2 (two) versions of every material, one using ARNE and the other using ALCO. The **creation** (Figure 7.6) and **modification** (Figure 7.7) tasks, which are related with the **learnability** evaluation, share a common structure, with 3 (three) Areas Of Interest (AOI): the *problem description* on the left-hand side; the *template key* below the problem description; and the *textual canvas* where participants would create or modify the use cases on the right hand-side.

The **understanding** (Figure 7.8) and **review** (Figure 7.9) tasks which are related with the **appropriateness recognisability** evaluation, share the same structure, with 2 (two) AOI: the *question* the participant is suppose to answer on top, and the *use case specification* about which the question is asked.

For each task, we used a similar layout with both templates, so that the only difference

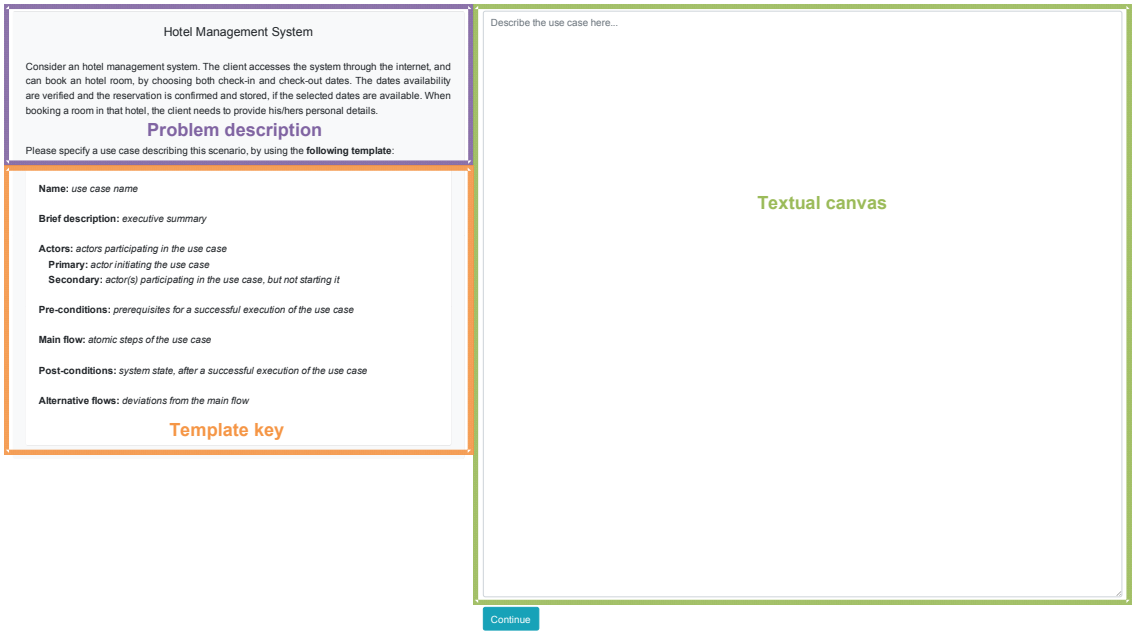


(a) Snapshot of the video tutorial for ARNE.

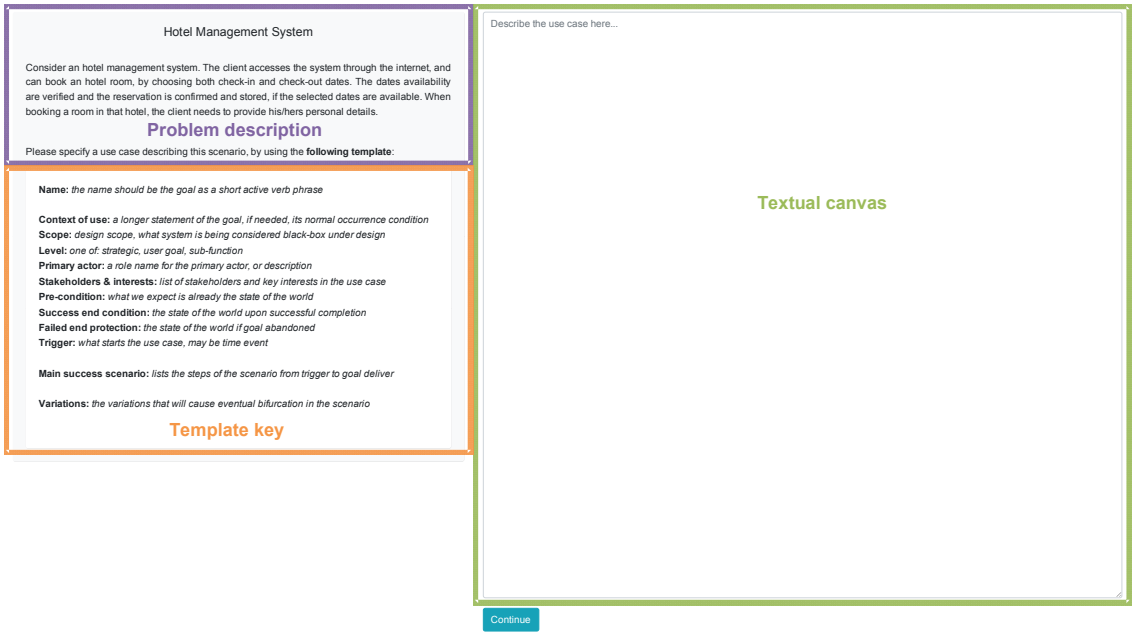


(b) Snapshot of the video tutorial for ALCO.

Figure 7.5: Snapshots of the use cases video tutorial viewed by the participants.

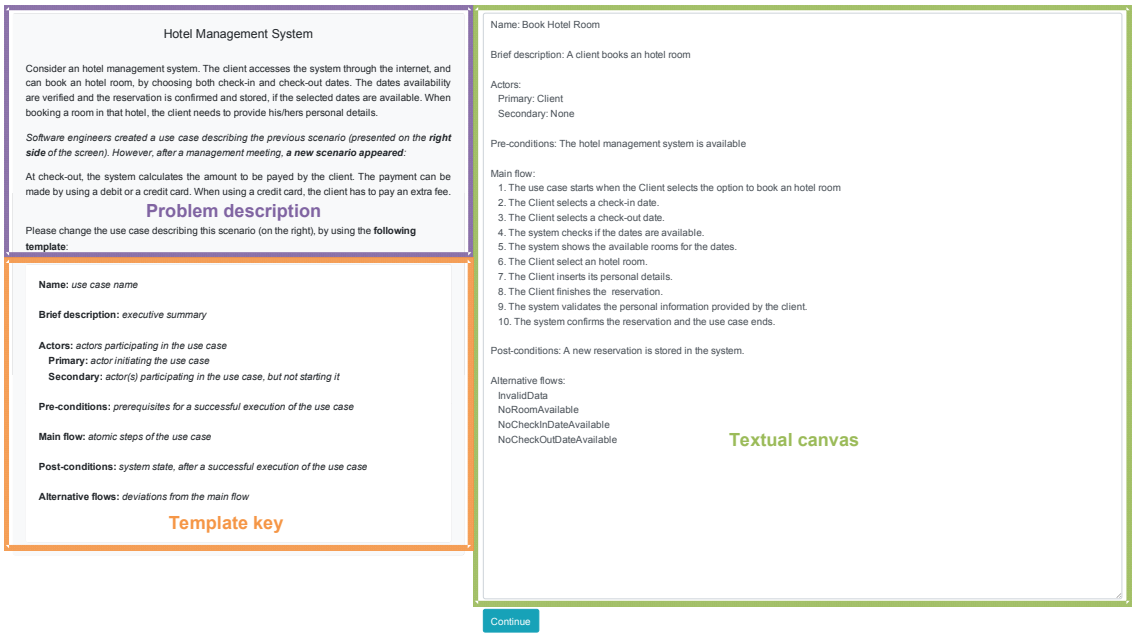


(a) AOI for the creation task with ARNE.

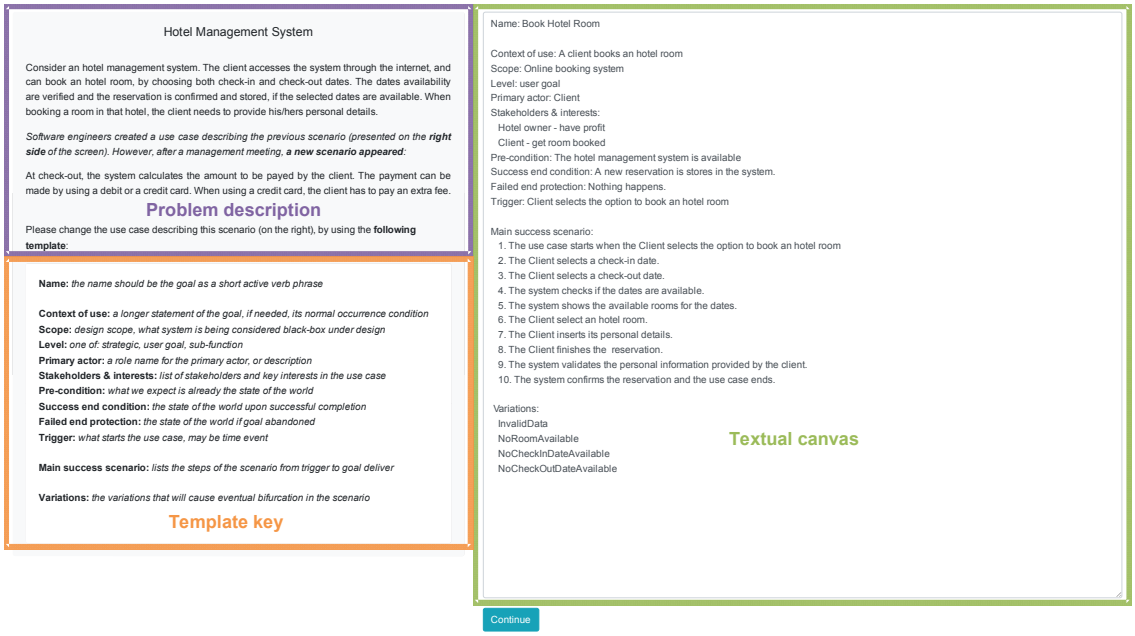


(b) AOI for the creation task with ALCO.

Figure 7.6: Creation tasks for the use case templates, illustrating the different AOI: the problem description on the left hand-side, the template key below the problem description, and the textual canvas on the remaining of the screen (with the initial use case specification for the modification task).

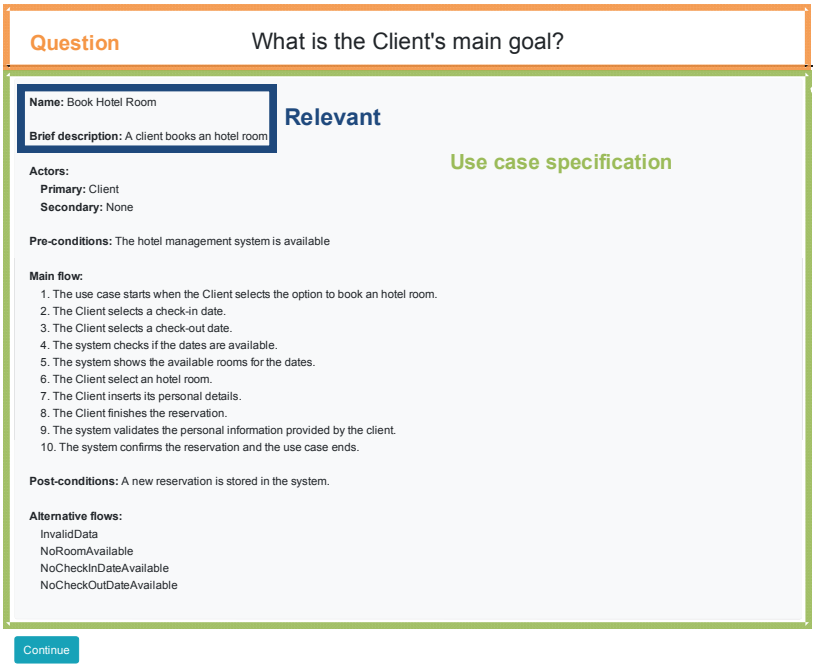


(a) AOI for the modification task with ARNE.

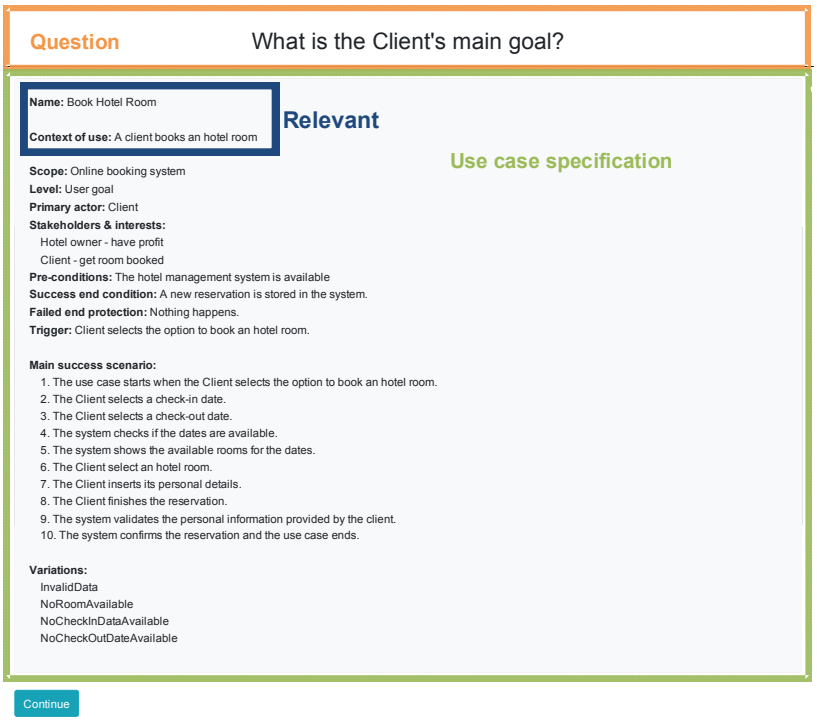


(b) AOI for the modification task with ALCO.

Figure 7.7: Modification task for the use case templates, illustrating the different AOI: the problem description on the left hand-side, the template key below the problem description, and the textual canvas on the remaining of the screen, with the initial use case specification.

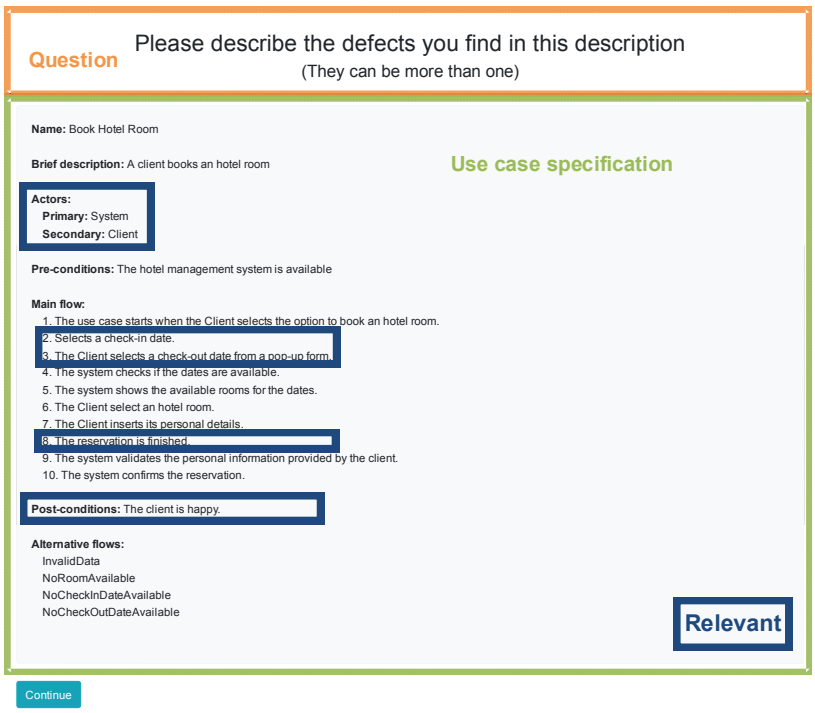


(a) AOI for the understanding task with ARNE.

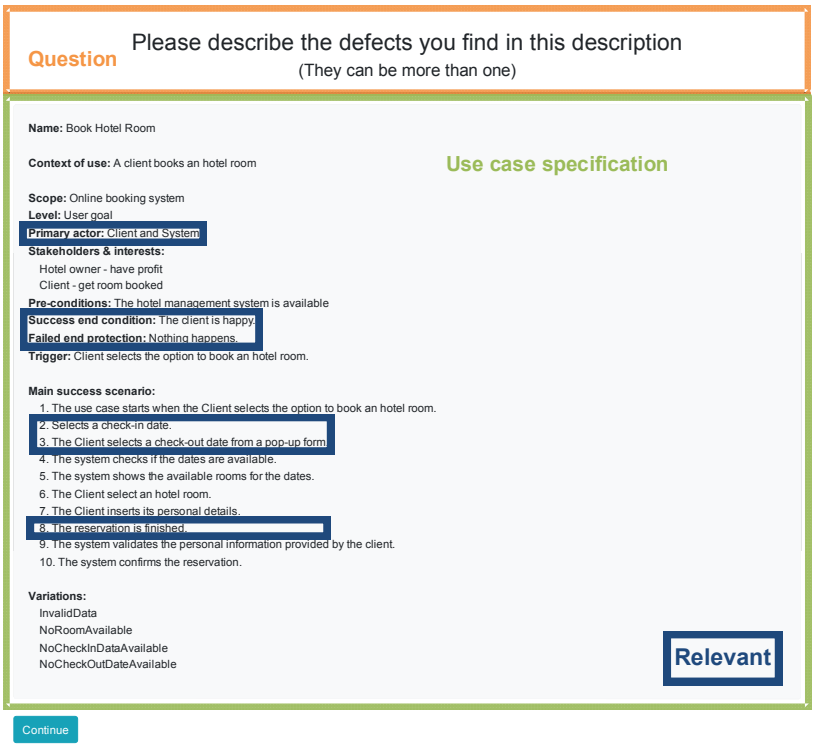


(b) AOI for the understanding task with ALCO.

Figure 7.8: Understanding tasks for the use case templates, illustrating the different AOI: the question on top, and the use case specification on the remaining of the screen.



(a) AOI for the review task with ARNE.



(b) AOI for the review task with ALCO.

Figure 7.9: Review tasks for the use case templates, illustrating the different AOI: the question on top, and the use case specification on the remaining of the screen.

among them is the use case template. For each task we further annotated 2 (two) sets of AOI to analyse eye-tracking data. An AOI is classified as *relevant* if it contains an element that belongs to the answer of the task, or *irrelevant* otherwise.

7.1.4 Tasks

For each use case template, there were 4 tasks: creation, modification, understanding and review.

In the **creation task**, participants had to create a use case specification, given a small problem description, as we illustrated in Figure 7.6a, for ARNE; and in Figure 7.6b for ALCO. In Text 5, we present the problem description for both use case templates.

Text 5: Problem description for the use case creation task

Consider an hotel management system. The client accesses the system through the internet, and can book an hotel room, by choosing both check-in and check-out dates. The dates availability are verified and the reservation is confirmed and stored, if the selected dates are available. When booking a room in that hotel, the client needs to provide his/hers personal details.

Please specify a use case describing this scenario, by using the ***following template****:*

[For ARNE template:]

Name: *use case name*

Brief description: *executive summary*

Actors: *actors participating in the use case*

Primary: *actor initiating the use case*

Secondary: *actor(s) participating in the use case, but not starting it*

Pre-conditions: *prerequisites for a successful execution of the use case*

Main flow: *atomic steps of the use case*

Post-conditions: *system state, after a successful execution of the use case*

Alternative flows: *deviations from the main flow*

[For ALCO template]

Name: *the name should be the goal as a short active verb phrase*

Context of use: *a longer statement of the goal, if needed, its normal occurrence condition*

Scope: *design scope, what system is being considered black-box under design*

Level*: one of: strategic, user goal, sub-function*

Primary actor: *a role name for the primary actor, or description*

Stakeholders & interests: *list of stakeholders and key interests in the use case*

Pre-condition: *what we expect is already the state of the world*

Success end condition: *the state of the world upon successful completion*

Failed end protection: *the state of the world if goal abandoned*
Trigger: *what starts the use case, may be time event*
Main success scenario: *lists the steps of the scenario from trigger to goal deliver*
Variations: *the variations that will cause eventual bifurcation in the scenario*

In the **modification task**, participants had to modify an initial use case specification, given a problem description and a new requirement, as we showed in Figure 7.7a, for ARNE; and in Figure 7.7b, for ALCO. In Text 6, we present the problem description and the new requirement, for both use case templates.

Text 6: Problem description for the use case modification task

Consider an hotel management system. The client accesses the system through the internet, and can book an hotel room, by choosing both check-in and check-out dates. The dates availability are verified and the reservation is confirmed and stored, if the selected dates are available. When booking a room in that hotel, the client needs to provide his/hers personal details.

Software engineers created a use case describing the previous scenario (presented on the ***right*** *side of the screen). However, after a management meeting,* ***a new scenario appeared****:*

At check-out, the system calculates the amount to be payed by the client. The payment can be made by using a debit or a credit card. When using a credit card, the client has to pay an extra fee.

Please change the use case describing this scenario (on the right) by using the ***following template****:*

[The same templates as in Text 5.]

In the **understanding task**, participants had to answer a total of 7 (seven) questions about a given use case specification, as we presented in Figure 7.8a, for ARNE; and in Figure 7.8b, for ALCO. The questions, appearing in a random order, aimed to cover the main tags of a use case specification. In Text 7, we present the questions, in no particular order.

Text 7: Set of questions for the use case understanding task

What is the Client's main goal?
Which information needs to be validated by the system?
What happens after a reservation is confirmed?
What can cause a reservation to not be successful?
What needs to happen before the Client starts the reservation?
Which actors are involved in selecting the check-in date?

'NoRoomAvailable' happens after which main step?

In the **reviewing task**, participants had to identify semantic defects on a given use case specification, as we illustrated in Figure 7.9a, for ARNE; and in Figure 7.9b, for ALCO. In Text 8, we present the assignment. We only informed the participants that their task was to find "defects". Explicitly describing the type of defects would have introduced a bias in the participants attention. This way, each participant was free to review the use case using his best judgment, as a real-world stakeholder would.

Text 8: Assignment for the use case reviewing task

Please describe the defects you find in this description
(They can be more than one)

7.1.5 Hypotheses, Parameters, and Variables

For each one of the goals presented in Subsection 7.1.1, we define the **null** (H_0) and **alternative hypotheses** (H_1). Following the same principle of the goals, all the hypotheses are similar, only changing the *underline and italic* part. However, they are fully specified for documentation purposes and easier reference.

The first set of hypotheses is related with the **use case templates** ($\boldsymbol{H_{0Tx}}$ and $\boldsymbol{H_{1Tx}}$) themselves, with the objective of comparing the differences between the results achieved when using ARNE and ALCO.

$\boldsymbol{H_{0T1}}$ Differences in the use case templates **do not** influence the *creation* of use case specifications.

- $\boldsymbol{H_{0T1.1}}$ Differences in the use case templates **do not** influence the *accuracy to create* use case specifications.
- $\boldsymbol{H_{0T1.2}}$ Differences in the use case templates **do not** influence the *speed to create* use case specifications.
- $\boldsymbol{H_{0T1.3}}$ Differences in the use case templates **do not** influence the *ease to create* use case specifications.

$\boldsymbol{H_{1T1}}$ Differences in the use case templates influence the *creation* of use case specifications.

- $\boldsymbol{H_{1T1.1}}$ Differences in the use case templates influence the *accuracy to create* use case specifications.
- $\boldsymbol{H_{1T1.2}}$ Differences in the use case templates influence the *speed to create* use case specifications.

$H_{1T1.3}$ Differences in the use case templates influence the *ease to create* use case specifications.

H_{0T2} Differences in the use case templates **do not** influence the *modification* of use case specifications.

$H_{0T2.1}$ Differences in the use case templates **do not** influence the *accuracy to modify* use case specifications.

$H_{0T2.2}$ Differences in the use case templates **do not** influence the *speed to modify* use case specifications.

$H_{0T2.3}$ Differences in the use case templates **do not** influence the *ease to modify* use case specifications.

H_{1T2} Differences in the use case templates influence the *modification* of use case specifications.

$H_{1T2.1}$ Differences in the use case templates influence the *accuracy to modify* use case specifications.

$H_{1T2.2}$ Differences in the use case templates influence the *speed to modify* use case specifications.

$H_{1T2.3}$ Differences in the use case templates influence the *ease to modify* use case specifications.

H_{0T3} Differences in the use case templates **do not** influence the *understanding* of use case specifications.

$H_{0T3.1}$ Differences in the use case templates **do not** influence the *accuracy to understand* use case specifications.

$H_{0T3.2}$ Differences in the use case templates **do not** influence the *speed to understand* use case specifications.

$H_{0T3.3}$ Differences in the use case templates **do not** influence the *ease to understand* use case specifications.

H_{1T3} Differences in the use case templates influence the *understanding* of use case specifications.

$H_{1T3.1}$ Differences in the use case templates influence the *accuracy to understand* use case specifications.

$H_{1T3.2}$ Differences in the use case templates influence the *speed to understand* use case specifications.

$H_{1T3.3}$ Differences in the use case templates influence the *ease to understand* use case specifications.

H_{0T4} Differences in the use case templates **do not** influence the *reviewing* of use case specifications.

- $H_{0T4.1}$ Differences in the use case templates **do not** influence the *accuracy to review* use case specifications.
- $H_{0T4.2}$ Differences in the use case templates **do not** influence the *speed to review* use case specifications.
- $H_{0T4.3}$ Differences in the use case templates **do not** influence the *ease to review* use case specifications.

H_{1T4} Differences in the use case templates influence the *reviewing* of use case specifications.

- $H_{1T4.1}$ Differences in the use case templates influence the *accuracy to review* use case specifications.
- $H_{1T4.2}$ Differences in the use case templates influence the *speed to review* use case specifications.
- $H_{1T4.3}$ Differences in the use case templates influence the *ease to review* use case specifications.

The second set of hypotheses is related with the levels of the GenderMag facets (H_{0GMx} and H_{1GMx}), with the objective of comparing the differences between the personas on each of the 5 (five) problem-solving facets.

H_{0GM1} Differences in the level of the GenderMag facets **do not** influence the *creation* of use case specifications.

- $H_{0GM1.1}$ Differences in the level of the GenderMag facets **do not** influence the *accuracy to create* use case specifications.
- $H_{0GM1.2}$ Differences in the level of the GenderMag facets **do not** influence the *speed to create* use case specifications.
- $H_{0GM1.3}$ Differences in the level of the GenderMag facets **do not** influence the *ease to create* use case specifications.

H_{1GM1} Differences in the level of the GenderMag facets influence the *creation* of *i** SR models.

- $H_{1GM1.1}$ Differences in the level of the GenderMag facets influence the *accuracy to create* use case specifications.
- $H_{1GM1.2}$ Differences in the level of the GenderMag facets influence the *speed to create* use case specifications.

$H_{1GM1.3}$ Differences in the level of the GenderMag facets influence the *ease to create* use case specifications.

H_{0GM2} Differences in the level of the GenderMag facets **do not** influence the *modification* of use case specifications.

$H_{0GM2.1}$ Differences in the level of the GenderMag facets **do not** influence the *accuracy to modify* use case specifications.

$H_{0GM2.2}$ Differences in the level of the GenderMag facets **do not** influence the *speed to modify* use case specifications.

$H_{0GM2.3}$ Differences in the level of the GenderMag facets **do not** influence the *ease to modify* use case specifications.

H_{1GM2} Differences in the level of the GenderMag facets influence the *modification* of use case specifications.

$H_{1GM2.1}$ Differences in the level of the GenderMag facets influence the *accuracy to modify* use case specifications.

$H_{1GM2.2}$ Differences in the level of the GenderMag facets influence the *speed to modify* use case specifications.

$H_{1GM2.3}$ Differences in the level of the GenderMag facets influence the *ease to modify* use case specifications.

H_{0GM3} Differences in the level of the GenderMag facets **do not** influence the *understanding* of use case specifications.

$H_{0GM3.1}$ Differences in the level of the GenderMag facets **do not** influence the *accuracy to understand* use case specifications.

$H_{0GM3.2}$ Differences in the level of the GenderMag facets **do not** influence the *speed to understand* use case specifications.

$H_{0GM3.3}$ Differences in the level of the GenderMag facets **do not** influence the *ease to understand* use case specifications.

H_{1GM3} Differences in the level of the GenderMag facets influence the *understanding* of use case specifications.

$H_{1GM3.1}$ Differences in the level of the GenderMag facets influence the *accuracy to understand* use case specifications.

$H_{1GM3.2}$ Differences in the level of the GenderMag facets influence the *speed to understand* use case specifications.

$H_{1GM3.3}$ Differences in the level of the GenderMag facets influence the *ease to understand* use case specifications.

H_{0GM4} Differences in the level of the GenderMag facets **do not** influence the *reviewing* of use case specifications.

- $H_{0GM4.1}$ Differences in the level of the GenderMag facets **do not** influence the *accuracy to review* use case specifications.
- $H_{0GM4.2}$ Differences in the level of the GenderMag facets **do not** influence the *speed to review* use case specifications.
- $H_{0GM4.3}$ Differences in the level of the GenderMag facets **do not** influence the *ease to review* use case specifications.

H_{1GM4} Differences in the level of the GenderMag facets influence the *reviewing* of use case specifications.

- $H_{1GM4.1}$ Differences in the level of the GenderMag facets influence the *accuracy to review* use case specifications.
- $H_{1GM4.2}$ Differences in the level of the GenderMag facets influence the *speed to review* use case specifications.
- $H_{1GM4.3}$ Differences in the level of the GenderMag facets influence the *ease to review* use case specifications.

For use case templates, the **independent variable** is the *template*, which may be ARNE, or ALCO. For *GenderMag*, the variable is the level of the facet – the *persona* – which may be Abby or Tim, on each of the 5 (five) facets (motivation for using software, information processing style, computer self-efficacy, attitude towards risk, and ways of learning new technology).

The **dependent variables** are *accuracy*, *speed*, *ease* (*visual*, *mental*, and *emotional*), and *perceived effort*. The variables and the corresponding metrics were fully described in Subsection 5.1.5.

7.1.6 Experimental Design

See Subsection 5.1.6.

7.2 Execution

7.2.1 Preparation

See Subsection 5.2.1.

7.2.2 Procedure

See Subsection 5.2.2.

7.2.3 Deviations from the Plan

See Subsection 5.2.3.

7.3 Analysis

7.3.1 Data Set Preparation

See Subsection 5.3.1.

7.3.2 Analysis Procedure

See Subsection 5.3.2.

7.3.3 Descriptive Statistics

In Table 7.1 we present the descriptive statistics for the metrics collected in our data analysis. For the sake of brevity, we only present the results concerning *accuracy* of *use case templates*, and including *precision*, *recall* and *f-measure*. Due to its high number, the remainder of the data can be found in a webpage [213].

For each metric, we present 8 lines in the Table. The first 2 refer to the creation task, the next 2 to the modification task, then 2 for the understanding task, and the last 2 are related with the review task. In the *Template* column we specify which of the use case templates we are considering: ARNE or ALCO. We further present the mean, standard deviation, skewness, kurtosis, and the p-value for the Shapiro-Wilk normality test. The shape of the distributions suggests that, in several cases, normality is **not** a reasonable assumption ($p < .05$). The variance of the distributions is not similar, for several of these variables.

The visual inspection of boxplot diagrams (in Figure 7.10),further reinforced our assessment concerning data normality.

7.3.4 Hypotheses Testing

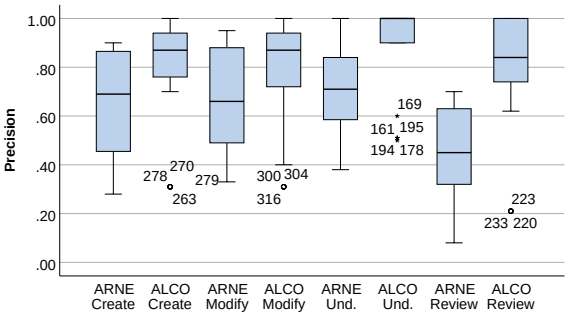
For testing our hypotheses, we used the *Welch's t-test*, as it is robust to deviations from the normal distribution, different sample sizes, and variance in the samples, thus following the recommendations on data analysis for Software Engineering empirical evaluations [121] (which summarises best practices in statistical analysis on other domains). We are using $p < .05$ for the level of significance and thus rejecting the null hypothesis.

In HGM_x, related with the levels of the GenderMag facets, we are also interested in comparing the different levels with the templates, so we used the *Factorial ANOVA test*.

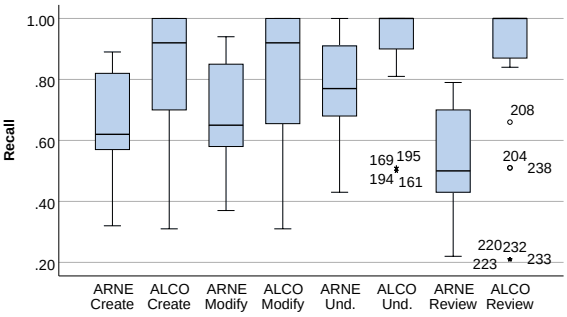
For the sake of brevity, we only present the results concerning RQT1, which serve to illustrate the results for the hypotheses testing. Due to its high number, the remainder of the data can be found in a webpage [213]. In this Section, we only present the results for the hypothesis testing. The discussion on the data can be found in Section 7.4.

Table 7.1: Descriptive statistics for *accuracy* when using ARNE and ALCO.

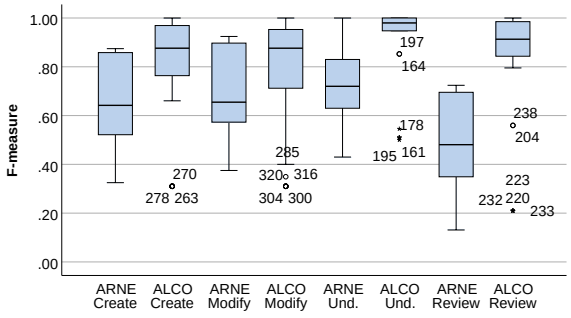
	Task	Template	Mean	S.D.	Skewness	Kurtosis	Shapiro-Wilk
Precision	Create	ARNE	.655	.212	-.413	-1.059	**.002**
		ALCO	.791	.238	-1.346	.411	**.000**
	Modify	ARNE	.672	.206	-.153	-1.055	**.006**
		ALCO	.780	.242	-1.177	-.072	**.000**
	Understand	ARNE	.713	.173	.027	-.670	.190
		ALCO	.916	.156	-2.062	2.971	**.000**
	Review	ARNE	.452	.200	-.339	-.987	**.006**
		ALCO	.802	.253	-1.544	1.420	**.000**
Recall	Create	ARNE	.658	.157	-.101	-.950	**.017**
		ALCO	.800	.257	-1.079	-.270	**.000**
	Modify	ARNE	.6830	.157	-.031	-.711	.055
		ALCO	.795	.264	-1.028	-.499	**.000**
	Understand	ARNE	.776	.152	-.280	-.784	.108
		ALCO	.916	.164	-2.011	2.602	**.000**
	Review	ARNE	.534	.157	.066	-.931	**.041**
		ALCO	.854	.274	-1.761	1.538	**.000**
F-measure	Create	ARNE	.650	.181	-.246	-1.086	**.001**
		ALCO	.793	.242	-1.301	.255	**.000**
	Modify	ARNE	.672	.178	-.039	-.980	**.002**
		ALCO	.784	.246	-1.197	-.132	**.000**
	Understand	ARNE	.738	.152	-.013	-.675	.231
		ALCO	.915	.157	-2.144	3.106	**.000**
	Review	ARNE	.478	.186	-.240	-.881	**.005**
		ALCO	.824	.255	-1.808	1.919	**.000**



(a) Precision for the use case templates.



(b) Recall for the use case templates.



(c) F-measure for the use case templates.

Figure 7.10: Boxplots for *accuracy* when using ARNE and ALCO.

RQT1: Does a difference in the use case templates (ARNE and ALCO) influence the ability to* create *use case specifications?

In Table 7.2 we summarise the Welch t-test results for the creation task, when comparing use case templates. There was a statistically significant difference in some of the variables ($p < .05$), with the p-value marked **bold** in the *Sig.* column of the Table.

Table 7.2: Welch t-test: *creation* task, use case templates.

	Metric	Statistic	df1	df2	Sig. (p-value)
Accuracy	Precision	7.256	1	76.932	**.009**
	Recall	8.843	1	64.476	**.004**
	F-measure	8.901	1	72.329	**.004**
Speed	Duration	72.819	1	75.458	**.000**
	FirstAct	11.743	1	77.998	**.001**
	LastAct	65.288	1	75.254	**.000**
	ProcDur	.257	1	77.552	.613
Visual ease	FixRel	1574.597	1	77.982	**.000**
	FixIrrel	176.672	1	39.035	**.000**
	AvgDurRelFix	23.082	1	77.744	**.000**
	AvgDurIrrelFix	40.587	1	77.106	**.000**
	TotSac	.264	1	77.315	.609
	Sac2Key	1.398	1	77.984	.241
Mental ease	AvgAttention	83.033	1	67.009	**.000**
	AvgMentWL	92.423	1	77.810	**.000**
	AvgFam	139.487	1	77.977	.000
Emot. ease	AvgSCL	2.042	1	76.900	.157
	AvgRMSSD	.109	1	77.938	.742
	AvgNN50	.005	1	77.996	.946
Perceived effort	Mental demand	.961	1	77.528	.330
	Physical demand	1.222	1	77.147	.272
	Temporal demand	19.557	1	76.540	**.000**
	Effort	.566	1	77.912	.454
	Performance	.567	1	70.996	.454
	Frustration	.246	1	78.000	.622
	NASA-TLX Score	.168	1	69.221	.684

Assessing accuracy. The *precision* achieved when using ARNE was *lower* ($M = .655$, $SD = .212$) than when using ALCO ($M = .791$, $SD = .238$, $t(1) = 7.256$, $p = .009$). Similarly, the *recall* achieved when using ARNE was *lower* ($M = .658$, $SD = .157$) than when using ALCO ($M = .800$, $SD = .257$, $t(1) = 8.843$, $p = .004$). Finally, the *f-measure* achieved when using ARNE was also *lower* ($M = .650$, $SD = .181$) than when using ALCO ($M = .793$, $SD = .241$, $t(1) = 8.901$, $p = .004$).

Assessing speed. The overall *duration* when using ARNE was *higher* ($M = 859.375$, $SD = 117.779$) than when using ALCO ($M = 652.800$, $SD = 97.821$, $t(1) = 72.819$, $p = .000$). However, the time for performing the *first action* when using ARNE was *lower*

($M = 149.175$, $SD = 58.873$) than when using ALCO ($M = 194.175$, $SD = 58.582$, $t(1) = 11.743$, $p = .001$). Finally, the time for performing the *last action* when using ARNE was *higher* ($M = 837.800$, $SD = 123.614$) than when using ALCO ($M = 633.150$, $SD = 101.879$, $t(1) = 65.288$, $p = .000$).

Assessing visual ease. The *fixation rate on relevant elements* when using ARNE was *higher* ($M = 10.205$, $SD = .724$) than when using ALCO ($M = 3.825$, $SD = .713$, $t(1) = 1574.597$, $p = .000$). Similarly, the *fixation rate on irrelevant elements* when using ARNE was *higher* ($M = 275.005$, $SD = 129.456$) than when using ALCO ($M = 2.877$, $SD = 2.742$, $t(1) = 176.672$, $p = .000$). On the other hand, the *average duration of relevant fixations* when using ARNE was *lower* ($M = 97.555$, $SD = 45.137$) than when using ALCO ($M = 147.500$, $SD = 47.808$, $t(1) = 23.082$, $p = .000$). Finally, the *average duration of irrelevant fixations* when using ARNE was also *lower* ($M = 226.369$, $SD = 128.730$) than when using ALCO ($M = 420.500$, $SD = 143.423$, $t(1) = 40.587$, $p = .000$).

Assessing mental ease. The *average attention* when using ARNE was *higher* ($M = .718$, $SD = .145$) than when using ALCO ($M = .335$, $SD = .223$, $t(1) = 82.033$, $p = .000$). In the same manner, the *average mental workload* when using ARNE was *higher* ($M = .733$, $SD = .172$) than when using ALCO ($M = .373$, $SD = .163$, $t(1) = 92.423$, $p = .000$).

Assessing emotional ease. For all the variables, we found no statistical evidence of differences between using ARNE and ALCO.

Assessing perceived effort. The perceived *temporal demand* when using ARNE was *higher* ($M = 34.000$, $SD = 16.992$) than when using ALCO ($M = 18.250$, $SD = 14.787$, $t(1) = 19.557$, $p = .000$).

7.4 Discussion

7.4.1 Evaluation of Results and Implications

RQT1: Does a difference in the use case templates (ARNE and ALCO) influence the ability to* create *use case specifications?

Assessing accuracy. There was a statistically significant difference between the templates in terms of accuracy. The *precision*, *recall* and *f-measure* achieved by participants using ARNE was lower than the one of those using ALCO. Although all the metrics were higher than 65%, for both templates, they increase to higher than 79% when using ALCO (almost 15 percentage points, more). With ALCO, the participants were able to create rather accurate use case specifications. These results were somewhat surprising, since ALCO is more detailed than ARNE, which could have caused the participants to become confused with the amount of information that needed to specified. However, that was not the case. We argue that the higher level of detail in the ALCO template made the participants to better analyse the problem description, which in turn allowed them to more easily understand what needed to be specified.

Assessing speed. There was a statistically significant difference between the templates in terms of *duration*. Participants using ARNE were ≈ 4 minutes slower to complete the task than the ones using ALCO. However, they started creating the model faster, taking less time to perform the *first action*. The time for performing the *last action* was higher for participants using ARNE, meaning they have spent more time actively creating the use case specification. Participants using ARNE started before, and they took more time to complete the task. Although a 4 minutes differences could be negligible, the task was to create a small use case specification. In a large software project, this difference in terms of duration can scale to a few hours. It would be acceptable if it translated into a more accurate specification. However, that time was not as effective as the (less) time used by the ALCO participants, as it can be seen in terms of *accuracy*. Nevertheless, we have to take into account the previous (lack of) knowledge of our participants. With training, we argue that the time to complete the task would decrease. Further studies are needed to better understand the learning curve of these templates.

Assessing visual ease. There was a statistically significant difference between the templates in terms of visual ease. Participants using ARNE had a greater visual effort than the ones using ALCO, observable through a higher *fixation rate on relevant elements* and *fixation rate on irrelevant elements*. However, the *average duration of relevant fixations* and the *average duration of irrelevant fixations* were lower. Our interpretation is that, although participants were looking at the right element, they were also looking at the other elements and having difficulties deciding which ones were relevant. Since participants created a wide range of different use case specifications, having an heat map representing the fixations and duration of those fixations would not provide us useful insights. As such, we decided not to create the heat map for this task.

Assessing mental ease. There was a statistically significant difference between the templates in terms of mental ease. Participants using ARNE had a higher *average attention* and *average mental workload* than the ones using ALCO. In both metrics, the results for ARNE were greater than 50%, which was not the case for ALCO. In fact, the participants had a difference of almost 40 percentage points in terms of *average attention* and *average mental workload*. Participants using ARNE were mentally engaged and attentive to the task they were performing. However, this higher mental effort has not translated into the creation of a more accurate use case specification, as it can be seen in terms of *accuracy*.

Assessing emotional ease. Although the mean for the *average skin conductive level* of participants using ARNE was slightly lower than the ones using ALCO; and the mean for the *heart rate variability*, for both RMSSD and NN50, of participants using ARNE was slightly higher than the ones using ALCO, the differences in the distributions are not statistically significant. We found no evidence on the impact of the template on the *emotional ease* of participants performing the creation task on a use case specification.

Assessing perceived effort. Although the mean for the perceived *mental demand* of participants using ARNE was slightly higher than in the ones using ALCO, the difference in the distribution is not statistically significant. Nevertheless, the perceived *mental*

demand was ≈ 50 (out of 100) for both templates. The participants perceived the task of creating a use case specification, independently of the template, as being somewhat mentally challenging, which is in line with the results obtained through the biometric data. There was a significant difference between the templates. The perceived *temporal demand* of participants using ARNE was higher than the one of those using ALCO. This is congruent with the results obtained in terms of *speed*, meaning the participants were aware of their time on the task.

RQT2: Does a difference in the use case templates (ARNE and ALCO) influence the ability to* modify *use case specifications?

Assessing accuracy. Although the mean for the *precision* of participants using ARNE was lower then the ones using ALCO, the differences in the distributions is not statistically significant. Nevertheless, ALCO has a higher *precision*, by more than 10 percentage points. There was a statistically significant difference between the templates in terms of *recall* and *f-measure*. The *recall* and *f-measure* achieved by participants using ARNE was lower than the one of those using ALCO. Although all the metrics were higher than 65%, for both templates, they increase to be higher than 78% when using ALCO (13 percentage points more). With ALCO, the participants were able to modify use case specification in a rather accurate fashion. These results were somewhat surprising, since ALCO is more detailed than ARNE, which could have caused the participants to become confused with the amount of information that needed to specified. However, that was not the case. We argue that the higher level of detail in the ALCO template made the participants to better analyse the problem description, which in turn allowed them to more easily understand what needed to be specified.

Assessing speed. There was a statistically significant difference between the templates in terms of *duration*. Participants using ARNE were ≈ 2 minutes slower to complete the task than the ones using ALCO. They also started creating the use case specification later on, taking more time to perform the *first action*. The time for performing the *last action* was also higher for participants using ARNE than for the ones using ALCO. Although a 2 minutes differences could be easily negligible, the task was to create a small use case specification. In a large software project, this difference in terms of duration can scale to a few hours. It would be acceptable if it translated into a more accurate specification. However, that time was not as effective as the (less) time used by the ALCO participants, as it can be seen in terms of *accuracy*. As in the creation task, we have to take into account the previous (lack of) knowledge of our participants. With training, we argue that the time to complete the task would decrease. Further studies are needed to better understand the learning curve of these templates.

Assessing visual ease. There was a statistically significant difference between the templates in terms of visual ease. Participants using ARNE had a greater visual effort than the ones using ALCO, observable through a higher *fixation rate on relevant elements* and

fixation rate on irrelevant elements. However, the *average duration of relevant fixations* and *average duration on irrelevant fixations* were lower. Our interpretation is that, although participants were looking at the right element, they were also looking at the other elements and having difficulties deciding which ones were relevant. Since participants modified the use case specification in a wide range of different ways, having an heat map representing the fixations and duration of those fixations would not provide us useful insights. As such, we decided not to create the heat map for this task.

Assessing mental ease. Although the mean for the *average attention* and *average mental workload* of participants using ARNE is higher than in the ones using ALCO, the differences in the distributions is not statistically significant. Nevertheless, these metrics were higher than 55% in both templates. Participants were somewhat mentally engaged and attentive to the task they were performing, independently of the template. There was one significant difference between the templates. Participants using ARNE had a higher *average familiarity* than the ones using ALCO. Nevertheless, the *average familiarity* was higher than 60% in both templates. Several participants said to have learnt use cases in the context of a course and some of them are still using them in a professional environment, hence the initial *familiarity* with the templates. However, this greater familiarity in the ARNE template had no impact on the *accuracy* of the task, nor on the *mental effort* while performing it.

Assessing emotional ease. Although the mean for the *average skin conductive level* of participants using ARNE was slightly lower than the ones using ALCO; and the mean for the *heart rate variability*, for both RMSSD and NN50, of participants using ARNE was slightly higher than the ones using ALCO, the differences in the distributions are not statistically significant. We found no evidence on the impact of the template on the *emotional ease* of participants performing the modification task on a use case specification.

Assessing perceived effort. There was no statistically significant difference between the templates, in any of the NASA-TLX components. We found no evidence on the impact of the template on the *perceived effort* of participants performing the modification task on the use case specification.

RQT3: Does a difference in the use case templates (ARNE and ALCO) influence the ability to* understand *use case specifications?

Assessing accuracy. There was a statistically significant difference between the templates in terms of accuracy. The *precision*, *recall* and *f-measure* achieved by participants using ARNE was lower than the one of those using ALCO. Although all the metrics were higher than 70%, for both templates, they increase to higher than 90% when using ALCO (more than 20 percentage points better). With ALCO, the participants were able to almost perfectly understand the use case specifications. These results were somewhat surprising, since ALCO is more detailed than ARNE, which could have caused the participants to become confused with the amount of information provided. However, that was not the

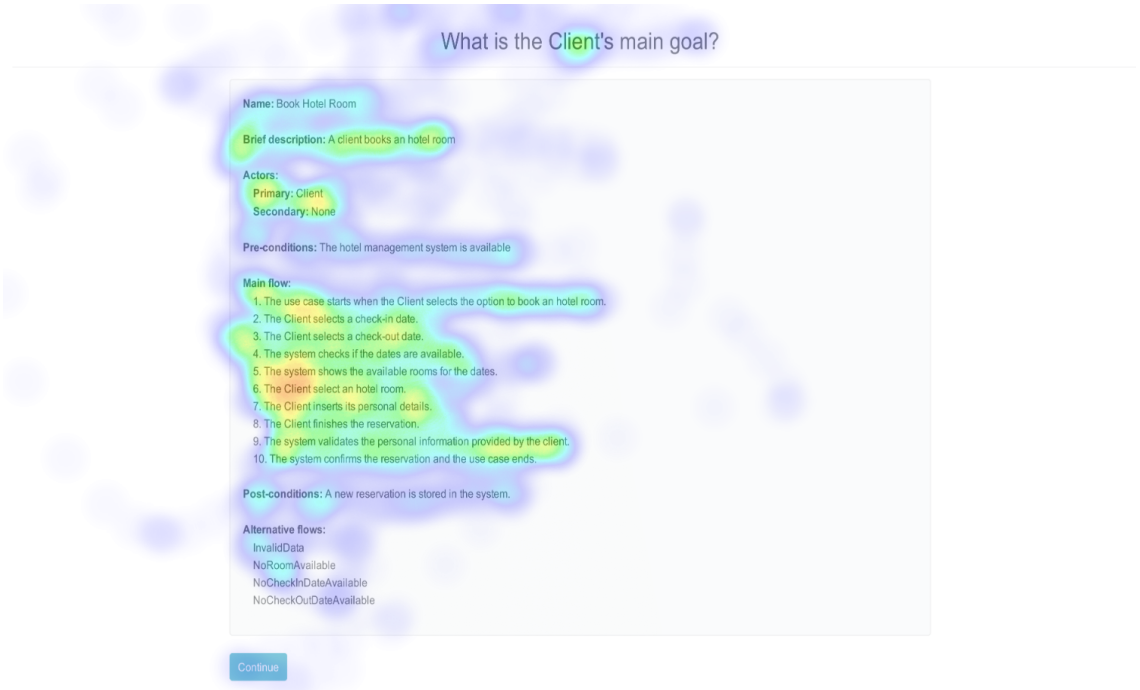
case. The higher level of detail in the ALCO template have not hindered the participants' ability to understand the use case. Nevertheless, *tacit knowledge* played a major role in the results, for both templates. When analysing the audio from the studies, the majority of the participants refer to their own experience when using an online booking system. Phrases like "*The industry pattern is to send an e-mail and show the information on the screen*" were common. Due to their experience with this type of systems, participants tended to use their knowledge instead of analysing the use case specification.

Assessing speed. The mean for *duration*, time for performing the *first detection*, and time for performing the *last detection* is highly similar in both templates, hence the differences in the distributions not being statistically significant. Nevertheless, the time was lower than 5 minutes. Participants were undoubtedly fast when understanding a use case specification, independently of the template. There was one significant difference between the templates. The *processing duration* was lower when using ARNE, meaning that, after replying to a question, the participant submits it without performing a thorough revision. We argue that this lack of a final analysis, per question, jeopardised the results.

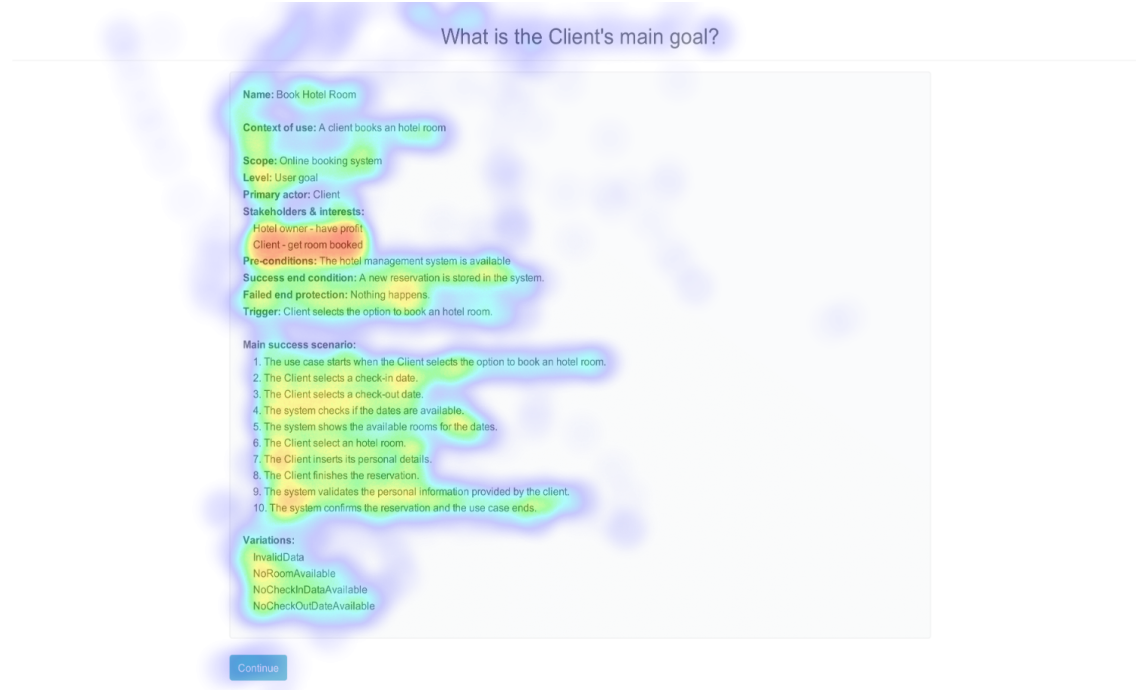
Assessing visual ease. There was a statistically significant difference between the templates in terms of visual ease. Participants using ARNE had a lower *fixation rate on relevant elements* and *average duration of irrelevant fixation* than the ones using ALCO. Our interpretation is that, although participants looked at the right element, they also spent some time analysing all the other elements available, but rapidly passing through the irrelevant ones. In Figure 7.11 we illustrate the heat maps representing the areas more frequently gazed during the understanding task, with ARNE, in Figure 7.11a; and ALCO, in Figure 7.11b. The heat maps further reinforce the conclusion that participants using ALCO had a greater fixation on relevant elements.

Assessing mental ease. Although the mean for the *average mental workload* of participants using ARNE was slightly higher than in the ones using ALCO, the difference in the distributions is not statistically significant. Nevertheless, the *average mental workload* was $\approx 50\%$ in both templates. Participants were somewhat mentally engaged on the task they were performing, independently of the template. There were some significant differences between the templates. The *average attention* was higher in participants using ARNE than in the ones using ALCO, indicating a greater attention and effort while performing the task. Participants using ARNE also have a higher *average familiarity* than the ones using ALCO. Some participants said to have learnt use cases in the context of a course and some of them are still using them in a professional environment, hence the initial *familiarity* with the templates. However, this greater familiarity in the ARNE template had no impact on the *accuracy* of the task, nor on the *mental effort* while performing it.

Assessing emotional ease. Although the mean for the *average skin conductive level* and *heart rate variability*, for both RMSSD and NN50, of participants using ARNE was higher than the ones using ALCO, the differences in the distributions are not statistically significant. We found no evidence on the impact of the template on the *emotional ease* of



(a) Heat map for ARNE understanding task.



(b) Heat map for ALCO understanding task.

Figure 7.11: Heat maps for fixations during use cases understanding task.

participants performing the understanding task on an use case specification.

Assessing perceived effort. There was no statistically significant difference between the templates, in any of the NASA-TLX components. We found no evidence on the impact of the template on the *perceived effort* of participants performing the understanding task on the use case specification.

RQT4: Does a difference in the use case templates (ARNE and ALCO) influence the ability to* review *use case specifications?

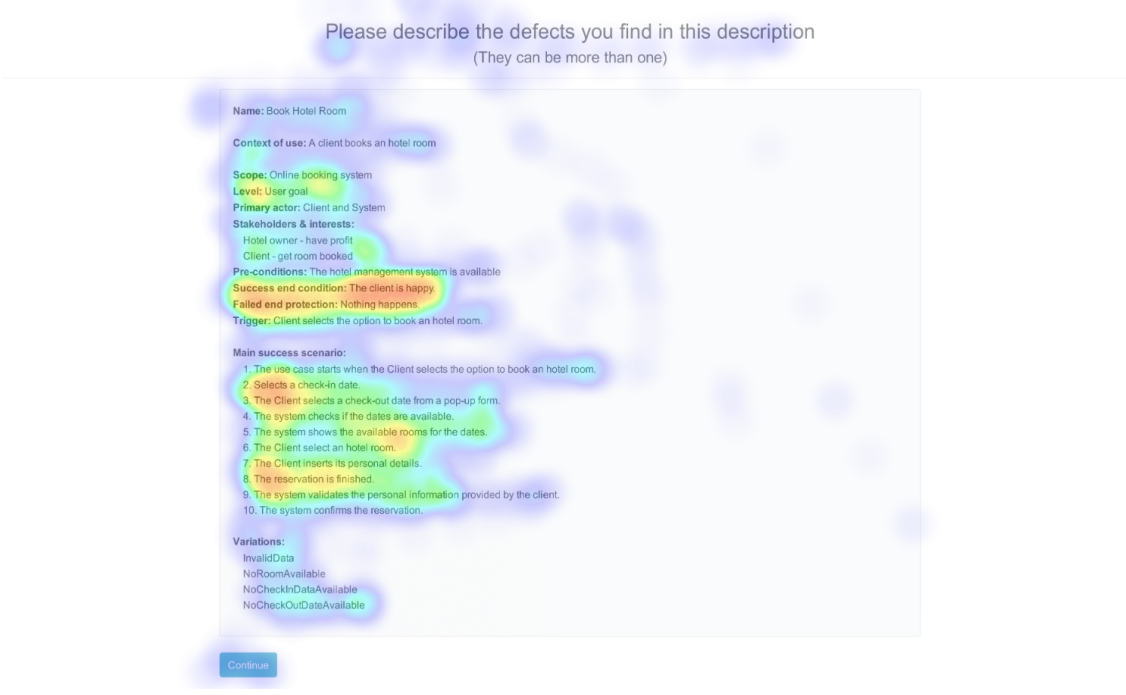
Assessing accuracy. There was a statistically significant difference between the templates in terms of accuracy. The *precision*, *recall* and *f-measure* achieved by participants using ARNE was lower than the one of those using ALCO. The results for ARNE were not great, being lower than 50% for both *precision* and *f-measure*. On the other hand, the results for participants using ALCO were higher than 80% (more than 30 percentage points better). Our participants really struggled when reviewing the use case with the ARNE template. These results were somewhat surprising, since ALCO is more detailed than ARNE, which could have caused the participants to become confused with the amount of information provided. One possible explanation is that, since it has a lower number of concepts, ARNE can encapsulate different information in the same concept, which might have made it harder for participants to identify the problems.

Assessing speed. There was a statistically significant difference between the templates in terms of *duration*. Participants using ARNE were ≈ 8 minutes slower to complete the task than the ones using ALCO. However, they started creating the model faster, taking less time to perform the *first action*. The time for performing the *last action* was higher for participants using ARNE, meaning they have spent more time actively creating the use case specification. Participants using ARNE started before, and they took more time to complete the task. An 8 minutes difference is somewhat expressive, specially when considering that the task was to review a small use case specification. Of course, this time can be reduced with training and experience, and it would be acceptable if it translated into a more accurate specification. However, that time was not as effective as the (less) time used by the ALCO participants, as it can be seen in terms of *accuracy*.

Assessing visual ease. There was a statistically significant difference between the templates in terms of visual ease. Participants using ARNE had a greater visual effort than the ones using ALCO, observable through a higher *fixation rate on relevant elements* and *fixation rate on irrelevant elements*. However, the *average duration of relevant fixations* and *average duration on irrelevant fixations* were lower. Our interpretation is that, although participants were looking at the right element, they were also looking at the other elements and having difficulties deciding which ones were relevant. In Figure 7.11 we illustrate the heat maps representing the areas more frequently gazed during the understanding task, with ARNE, in Figure 7.11a; and ALCO, in Figure 7.11b. The heat maps further

reinforce the conclusion that participants using ALCO had a greater visual effort than the ones using ARNE.

(a) Heat map for ARNE review task.



(b) Heat map for ALCO review task.

Figure 7.12: Heat maps for fixations during use cases review task.

Assessing mental ease. There was a statistically significant difference between the templates in terms of mental ease. Participants using ARNE had a higher *average attention*,

average mental workload, and *average familiarity* than the ones using ALCO. Participants using ARNE were highly mentally engaged and attentive to the task they were performing. However, this higher mental effort has not translated into a more accurate review of use case specification, as it can be seen in terms of *accuracy*. Further, the greater familiarity in the ARNE template had no impact on the *accuracy* of the task, nor on the *mental effort* while performing it.

Assessing emotional ease. Although the mean for the *average skin conductive level* and *heart rate variability*, for both RMSSD and NN50, of participants using ARNE was slightly lower than the ones using ALCO, the differences in the distributions are not statistically significant. We found no evidence on the impact of the template on the *emotional ease* of participants performing the reviewing task on a use case specification.

Assessing perceived effort. There was a significant difference between the templates in terms of perceived effort. The perceived *mental demand* of participants using ARNE was higher than the one of those using ALCO. The result was higher than 50 (out of 100) for both templates. The participants perceived the task of reviewing a use case specification as being somewhat mentally challenging, which is in line with the results obtained through the biometric data. Furthermore, the perceived *temporal demand* of participants using ARNE was higher than the one of those using ALCO. This is congruent with the results obtained in terms of *speed*, meaning the participants were aware of their time on the task.

RQGM1: Does a difference in the level (Abby and Tim) of each GenderMag facet influence the ability to* create *use case specifications?

We found no evidence that the *motivation*, *self-efficacy*, and *learning style* facets influence the accuracy, speed, ease (visual, mental, and emotional), and perceived effort when performing the creation task on use case specifications.

Assessing accuracy. There was a statistically significant difference between Abby and Tim in one of the facets, in terms of accuracy. Participants characterised as Abby in the *risk* facet had a higher *precision*, when compared with those identified as Tim. Abby is risk-averse and only answers when she's sure. As such, when she answers, her answer tends to be correct.

Assessing speed. There was a statistically significant difference between Abby and Tim in some of the facets, in terms of speed. Participants characterised as Tim in the *information processing* facet had a higher *processing duration* than participants identified as Abby. This was somewhat surprising, since Abby is generally more comprehensive when analysing information, and typically prefers to revise the performed task to make sure that nothing was forgotten. However, the difference was of only 9 seconds, which can be negligible. As such, their processing duration can be considered similar. Tim in the *risk* facet makes the *first action* in the use case specification really early. In fact, the eye-tracking data shows he starts trying to solve the task even before finishing reading

the problem description. As for Abby, she only starts after some time. Plus, the *processing duration* was lower for Tim. This means that, after the creation of the use case specification, Tim submits it without performing a revision. We argue that this is due to his high confidence on his work.

Assessing visual ease. There was a statistically significant difference between Abby and Tim in some of the facets, in terms of visual ease. Participants characterised as Abby in the *information processing* and *risk* facet had a greater visual effort, observable through a higher *average duration on irrelevant fixations*. However, Abby had a lower *average duration of relevant fixations*. Our interpretation is that Tim, being more selective in the way he processes information, is able to focus more on the relevant elements. As for Abby, she tends to further analyse the information provided, hence the focus on the irrelevant elements.

Assessing mental ease. There was a statistically significant difference between Abby and Tim in some of the facets, in terms of mental ease. Participants characterised as Abby in the *information processing* facet had a greater mental effort, observable through a higher *average attention* and *average mental workload*. Since Abby is more comprehensive when processing information, her level of attention indicates she is highly engaged and attentive to the task she is performing. On the other hand, participants characterised as Tim in the *risk* facet had a higher *mental workload*. This was somewhat surprising and further studies are needed to understand why.

Assessing emotional ease. There was a statistically significant difference between Abby and Tim in the *risk* facet, in terms of emotional ease. Participants characterised as Tim had a higher *heart rate variability*, for both *RMSSD* and *NN50*, than the ones identified as Abby. Since Tim is risk-tolerant, our interpretation is that Tim was excited when performing the task. On the other hand, participants identified as Abby in the *information processing* facet had a higher *heart rate variability*, for *RMSSD*, than the ones characterised as Tim. Given that Abby is more comprehensive when processing information, we argue that the amount of information in the problem description might have made her to feel more stressed and anxious.

Assessing perceived effort. There was a statistically significant difference between Abby and Tim in some of the facets, in terms of perceived effort. Participants characterised as Abby in the *information processing* and *risk* facets had a higher perceived *mental demand* than the ones characterised as Tim. Finally, the *frustration* and *temporal demand* was higher for Abby in the *risk* facet. This is in line with the results obtained in terms of speed and through the biometric data, meaning the participants were well aware of their effort on the task.

RQGM2: Does a difference in the level (Abby and Tim) of each GenderMag facet influence the ability to* modify *use case specifications?

Assessing accuracy. There was a statistically significant difference between Abby and Tim in some of the facets, in terms of accuracy. Participants characterised as Abby in the *self-efficacy* and *risk* facets had a higher *precision*, when compared with those identified as Tim. Furthermore, Abby in these facets also had a higher *recall*. Our interpretation is that having an initial use case specification, helped Abby to better understand the templates and more effectively modify the use case specification.

Assessing speed. There was a statistically significant difference between Abby and Tim in some of the facets, in terms of speed. Participants characterised as Tim in the *information processing* facet had a higher *duration* and *processing duration* than participants identified as Abby. This was somewhat surprising, since Abby is generally more comprehensive when analysing information, and typically prefers to revise the performed task to make sure that nothing was forgotten. However, the difference was lower than 1 minute, which can be considered negligible. Tim in the *risk* facet makes the *first action* in the use case specification really early. In fact, he starts trying to solve the task even before finishing reading the problem description. As for Abby, she only starts after some time. Furthermore, the *processing duration* was lower for Tim, being less than 5 seconds. This means that, after the creation of the use case specification, Tim submits it without performing a revision. We argue that this is due to his high confidence on his work.

Assessing visual ease. There was a statistically significant difference between Abby and Tim in some of the facets, in terms of visual ease. Participants characterised as Abby in the *information processing*, *learning style* and *risk* facets had a greater visual effort, observable through a higher *fixation rate on irrelevant elements* and *average duration of irrelevant fixations*. Abby tends to further analyse the information provided, hence the focus on the irrelevant elements. However, it may also be the case that Abby is focused on the irrelevant elements but not being able to completely understand they are irrelevant.

Assessing mental ease. There was a statistically significant difference between Abby and Tim in some of the facets, in terms of mental ease. Participants characterised as Abby in the *information processing* and *self-efficacy* facets had a greater mental effort, observable through a higher *average attention* and *average mental workload*. Since Abby is more comprehensive when processing information, her level of attention indicates she is highly engaged and attentive to the task she is performing. On the other hand, participants characterised as Tim in the *risk* facet had a higher *mental workload*. This was somewhat surprising and further studies are needed to understand why.

Assessing emotional ease. There was a statistically significant difference between Abby and Tim in some of the facets, in terms of emotional ease. Participants characterised as Tim in the *motivation* and *risk* facet had a higher *heart rate variability*, for both *RMSSD* and *NN50*, than the ones identified as Abby. Since Tim is risk-tolerant and sees technology as a source of fun, our interpretation is that Tim was excited when performing the task. On the other hand, participants identified as Abby in the *information processing* facet had a higher *average skin conductive level* and *heart rate variability*, for *RMSSD*, than the ones characterised as Tim. Given that Abby is more comprehensive when processing

information, we argue that the amount of information in the problem description and in the initial use case specification might have made her to feel more stressed and anxious.

Assessing perceived effort. There was a statistically significant difference between Abby and Tim in some of the facets, in terms of perceived effort. Participants characterised as Abby in the *information processing* facet had a higher perceived *mental demand* than the ones characterised as Tim. This is in line with the results obtained through the biometric data. For that same facet, the perceived *temporal demand* of Abby was also higher. However, this is not observable in terms of the duration of the task. Our interpretation is that, since Abby *mental workload* was higher, this might have affected her ability to assess time, and the task seemed longer than it was. Finally, the *frustration* was higher for Abby in the *risk* facet. This is in line with the results obtained in terms of through the biometric data, meaning the participants were well aware of their effort on the task.

RQGM3: Does a difference in the level (Abby and Tim) of each GenderMag facet influence the ability to* understand *use case specifications?

We found no evidence that the *learning style* facets influence the accuracy, speed, ease (visual, mental, and emotional), and perceived effort when performing the understanding task on use case specifications.

Assessing accuracy. There was a statistically significant difference between Abby and Tim in one of the facets, in terms of accuracy. Participants characterised as Abby in the *risk* facet had a higher *precision*, when compared with those identified as Tim. Abby is risk-averse and only answers when she is sure. As such, when she answers, her answer tends to be correct. Furthermore, Abby in this facet also had a higher *recall*. Tim tented to use his tacit knowledge to answer the questions, which may have compromised the results, both in terms of accuracy and recall.

Assessing speed. There was a statistically significant difference between Abby and Tim in some of the facets, in terms of speed. Participants characterised as Tim in the *information processing* facet had a higher *duration* and *processing duration* than participants identified as Abby. This was somewhat surprising, since Abby is generally more comprehensive when analysing information, and typically prefers to revise the performed task to make sure that nothing was forgotten. However, the difference was of only 12 seconds, which can be negligible. As such, their processing duration can be considered similar. Tim in the *risk* facet makes the *first detection* in the use case specification really early. In fact, he starts trying to solve the task even before finishing reading the available use case specification. As for Abby, she only starts after some time.

Assessing visual ease. There was a statistically significant difference between Abby and Tim in some of the facets, in terms of visual ease. Participants characterised as Abby in the *information processing* and *risk* facet had a greater visual effort, observable through a higher *average duration on irrelevant fixations*. However, Abby had a lower *average duration of relevant fixations*. Our interpretation is that Tim, being more selective in the way he

processes information, is able to focus more on the relevant elements. As for Abby, she tends to further analyse the information provided, hence the focus on the irrelevant elements.

Assessing mental ease. There was a statistically significant difference between Abby and Tim in some of the facets, in terms of mental ease. Participants characterised as Abby in the *information processing* facet had a greater mental effort, observable through a higher *average attention* and *average mental workload*. Since Abby is more comprehensive when processing information, her level of attention indicates she is highly engaged and attentive to the task she is performing. On the other hand, participants characterised as Tim in the *risk* facet had a higher *mental workload*. This was somewhat surprising and further studies are needed to understand why.

Assessing emotional ease. There was a statistically significant difference between Abby and Tim in some of the facets, in terms of emotional ease. Participants characterised as Tim in the *motivation* and *risk* facet had a higher *heart rate variability*, for both *RMSSD* and *NN50*, than the ones identified as Abby. Since Tim is risk-tolerant and sees technology as a source of fun, our interpretation is that Tim was excited when performing the task. On the other hand, participants identified as Abby in the *information processing* facet had a higher *heart rate variability*, for *RMSSD*, than the ones characterised as Tim. Given that Abby is more comprehensive when processing information, we argue that the amount of information in the use case specification might have made her to feel more stressed.

Assessing perceived effort. There was a statistically significant difference between Abby and Tim in some of the facets, in terms of perceived effort. Participants characterised as Abby in the *self-efficacy* and *risk* facets had a higher perceived *temporal demand* than the ones identified as Tim. However, this is not observable in terms of the duration of the task. The *frustration* was also higher for Abby in these facets, which may have affected her ability to assess time, and the task seemed longer than it was.

RQGM4: Does a difference in the level (Abby and Tim) of each GenderMag facet influence the ability to* review *use case specifications?

Assessing accuracy. There was a statistically significant difference between Abby and Tim in some of the facets, in terms of accuracy. Participants characterised as Abby in the *self-efficacy* and *risk* facets had a higher *precision*, when compared with those identified as Tim. Abby is risk-averse and only answers when she is sure. As such, when she answers, her answer tends to be correct.

Assessing speed. There was a statistically significant difference between Abby and Tim in some of the facets, in terms of speed. Participants characterised as Tim in the *information processing* facet had a higher *processing duration* than participants identified as Abby. This was somewhat surprising, since Abby is generally more comprehensive when analysing information, and typically prefers to revise the performed task to make sure that nothing was forgotten. However, the difference was of only 7 seconds, which can

be negligible. As such, their *processing duration* can be considered similar. Tim in the *risk* facet makes the *first detection* in the use case specification really early. In fact, he starts trying to solve the task even before finishing reading the available use case specification. As for Abby, she only starts after some time.

Assessing visual ease. There was a statistically significant difference between Abby and Tim in some of the facets, in terms of visual ease. Participants characterised as Abby in the *information processing* and *motivation* facet had a greater visual effort, observable through a higher *average duration on irrelevant fixations*. However, Abby had a lower *average duration of relevant fixations*. Our interpretation is that Tim, being more selective in the way he processes information, is able to focus more on the relevant elements. As for Abby, she tends to further analyse the information provided, hence the focus on the irrelevant elements. On the other hand, participants identified as Tim in the *self-efficacy*, *motivation*, and *learning style* and a higher *total number of saccades*. Since Tim tends to have a tinkering approach, he's experimenting and changing his focus quickly.

Assessing mental ease. There was a statistically significant difference between Abby and Tim in some of the facets, in terms of mental ease. Participants characterised as Abby in the *information processing*, *learning style*, and *self-efficacy* facets had a greater mental effort, observable through a higher *average attention* and *average mental workload*. Since Abby is more comprehensive when processing information, her level of attention indicates she is highly engaged in the task she is performing.

Assessing emotional ease. There was a statistically significant difference between Abby and Tim in some of the facets, in terms of emotional ease. Participants characterised as Tim in the *risk* facet had a higher *heart rate variability*, for both *RMSSD* and *NN50*, than the ones identified as Abby. Since Tim is risk-tolerant, our interpretation is that Tim was excited when performing the task. On the other hand, participants identified as Abby in the *information processing* facet had a higher *heart rate variability*, for *RMSSD*, than the ones characterised as Tim. Given that Abby is more comprehensive when processing information, we argue that the amount of information in the use case specification might have made her to feel more stressed.

Assessing perceived effort. There was a statistically significant difference between Abby and Tim in some of the facets, in terms of perceived effort. There was a statistically significant difference between Abby and Tim in some of the facets, in terms of perceived effort. Participants characterised as Abby in the *informatiton processing* and *risk* facets had a higher perceived *temporal demand* than the ones identified as Tim. However, this is not observable in terms of the duration of the task. The *frustration* was also higher for Abby in these facets, which may have affected her ability to assess time, and the task seemed longer than it was.

7.4.2 Inferences

The ALCO template outperformed ARNE. For the majority of the metrics, participants were able to achieve a better performance and lower effort when using ALCO than the participants using ARNE. These results were somewhat surprising, since ALCO is more detailed than ARNE, which could have caused the participants to become confused or apprehensive with the amount of information. However, that was not the case.

Tacit knowledge may hinder the performance of the understanding task. When analysing the replies of our participants to the understanding task, the majority of them refers to their own experience when using an online booking system. Phrases like *"The industry pattern is to send an e-mail and show the information on the screen"*, *"It depends if a deposit was made or not"*, or *"I normally pay with debit card."* were quite common. Due to their experience with this type of systems, participants tended to use their knowledge instead of analysing the use case specification, and that made them ignore what was written in the use case specification and impaired their performance. This was particularly problematic in the understanding task.

Risk has impact on accuracy. Participants identified as Abby in this facets were able to achieve a good level of precision, even when it was their first contact with use cases. However, her attitude towards risk is sometimes undermining the recall. We argue that, with training, Abby would become more confident in her skills and could achieve great results for both precision and recall. As for Tim, making him aware that risking too much is possibly sabotaging his results could help with his precision.

Risk has impact on speed. Participants characterised as Tim in this facet tend to start trying to solve the task even before finishing reading the problem description. Moreover, he submits the answer without any further revision. By not revising the model, Tim may be losing an opportunity for improvement of his answer and for a higher precision.

Information processing has impact on ease. Abby in this facet has a more comprehensive analysis of the problem description and the use case specification. In particular, the mental workload is higher due to this thorough inspection. Plus, in general, Abby is more engaged at the task she's performing. Tim, however, is able to better separate what is relevant from what is not. We argue that, in this particular scenario, having a higher effort is not perceived as being harmful. Nonetheless, being able to more precisely understand what is relevant is a great advantage in terms of effort.

People diversity is key. When analysing the GenderMag result, we note that complementarity of the results achieved by Tim and Abby suggests that, rather than targeting the requirements process to one of them, there is more to be gained in leveraging their diversity. One possible way of doing so would be to build up teams with this diversity, specially in terms of *risk*.

7.5 Summary

We performed a family of quasi-experiments to analyse the impact of different use case templates, as well as different levels in each of the five GenderMag facets, when creating, modifying, understanding and reviewing use case specifications. We measured the accuracy, speed, ease (visual, mental, and emotional), and perceived effort of a total of 320 participants. We used metrics of task success, time, and effort, collected with eye-tracking, EEG and EDA sensors, and participants' feedback through a NASA-TLX questionnaire.

We found that our participants were able to achieve better results with ALCO than with ARNE. However, tacit knowledge had a great impact on the overall performance, specially in the understanding task. Furthermore, there are several differences in the individual characteristics (the GenderMag facets) of the participants that had an influence on their performance and effort.

Chapter 8

Comparison of *i** and Use Cases

In this Chapter, we present a comparison of *i** 1.0, iStar 2.0, ARNE use case template, and ARCO use case template. This comparison is based on the data from the quasi-experiments reported in Chapters 6 and 7. As such, and although following Jedlitschka *et al.* guidelines [113] on how to report (quasi-)experiments in Software Engineering, we have omitted both experiment planning (except for goals and hypotheses) and execution from this Chapter, given they are described in the previous Chapters.

8.1 Experiments Planning

8.1.1 Goals

We describe our research goals using the GQM research goal template [7, 8]. We analyse differences in 2 (two) main sets, related with: requirements languages, and levels of the GenderMag facets. Each set has 4 (four) main goals, each related with the tasks performed by the participants: creation, modification, understanding, and review. Finally, each high level goal has a set of sub-goals, related with accuracy, speed, and ease, which are also defined. All the goals are similar, only changing the *underline and italic* part. However, they are fully specified for documentation purposes and easier reference.

The first set of goals is related with the **requirements languages** (**GL**) themselves. The objective is to compare the differences between the results achieved when using *i** 1.0, iStar 2.0, ARNE use case template, and ALCO use case template.

(GL1) **Analyse** differences in the requirements language, **for the purpose of** evaluation, **with respect to** their effects on the *creation* of requirements models, **from the viewpoint of** researchers, **in the context of** experiments conducted at our University and at software companies.

(GL1.1) Analyse differences in the requirements language, **for the purpose of** evaluation, **with respect to** their effects on the *accuracy to create* requirements models, **from the viewpoint of** researchers, **in the context of** experiments conducted at our University and at software companies.

(GL1.2) Analyse differences in the requirements language, **for the purpose of** evaluation, **with respect to** their effects on the *speed to create* requirements models, **from the viewpoint of** researchers, **in the context of** experiments conducted at our University and at software companies.

(GL1.3) Analyse differences in the requirements language, **for the purpose of** evaluation, **with respect to** their effects on the *ease to create* requirements models, **from the viewpoint of** researchers, **in the context of** experiments conducted at our University and at software companies.

(GL2) Analyse differences in the requirements language, **for the purpose of** evaluation, **with respect to** their effects on the *modification* of requirements models, **from the viewpoint of** researchers, **in the context of** experiments conducted at our University and at software companies.

(GL2.1) Analyse differences in the requirements language, **for the purpose of** evaluation, **with respect to** their effects on the *accuracy to modify* requirements models **from the viewpoint of** researchers, **in the context of** experiments conducted at our University and at software companies.

(GL2.2) Analyse differences in the requirements language, **for the purpose of** evaluation, **with respect to** their effects on the *speed to modify* requirements models, **from the viewpoint of** researchers, **in the context of** experiments conducted at our University and at software companies.

(GL2.3) Analyse differences in the requirements language, **for the purpose of** evaluation, **with respect to** their effects on the *ease to modify* requirements models, **from the viewpoint of** researchers, **in the context of** experiments conducted at our University and at software companies.

(GL3) Analyse differences in the requirements language, **for the purpose of** evaluation, **with respect to** their effects on the *understanding* of requirements models, **from the viewpoint of** researchers, **in the context of** experiments conducted at our University and at software companies.

(GL3.1) Analyse differences in the requirements language, **for the purpose of** evaluation, **with respect to** their effects on the *accuracy to understand* requirements models, **from the viewpoint of** researchers, **in the context of** experiments conducted at our University and at software companies.

(GL3.2) Analyse differences in the requirements language, **for the purpose of** evaluation, **with respect to** their effects on the *speed to understand* requirements

models, **from the viewpoint of** researchers, **in the context of** experiments conducted at our University and at software companies.

(GL3.3) Analyse differences in the requirements language, **for the purpose of** evaluation, **with respect to** their effects on the *ease to understand* requirements models, **from the viewpoint of** researchers, **in the context of** experiments conducted at our University and at software companies.

(GL4) Analyse differences in the requirements language, **for the purpose of** evaluation, **with respect to** their effects on the *reviewing* of requirements models, **from the viewpoint of** researchers, **in the context of** experiments conducted at our University and at software companies.

(GL4.1) Analyse differences in the requirements language, **for the purpose of** evaluation, **with respect to** their effects on the *accuracy to review* requirements models, **from the viewpoint of** researchers, **in the context of** experiments conducted at our University and at software companies.

(GL4.2) Analyse differences in the requirements language, **for the purpose of** evaluation, **with respect to** their effects on the *speed to review* requirements models, **from the viewpoint of** researchers, **in the context of** experiments conducted at our University and at software companies.

(GL4.3) Analyse differences in the requirements language, **for the purpose of** evaluation, **with respect to** their effects on the *ease to review* requirements models, **from the viewpoint of** researchers, **in the context of** experiments conducted at our University and at software companies.

The second set of goals is related with the **levels of the GenderMag facets** (GGM). The objective is to compare the differences between the personas (Abby and Tim) on each of the 5 (five) facets (motivation for using software, information processing style, computer self-efficacy, attitude towards risk, and ways of learning new technology).

(GGM1) Analyse differences in the level of the GenderMag facets, **for the purpose of** evaluation, **with respect to** their effects on the *creation* of requirements models, **from the viewpoint of** researchers, **in the context of** experiments conducted at our University and at software companies.

(GGM1.1) Analyse differences in the level of the GenderMag facets, **for the purpose of** evaluation, **with respect to** their effects on the *accuracy to create* requirements models, **from the viewpoint of** researchers, **in the context of** experiments conducted at our University and at software companies.

(GGM1.2) Analyse differences in the level of the GenderMag facets, **for the purpose of** evaluation, **with respect to** their effects on the *speed to create* requirements models, **from the viewpoint of** researchers, **in the context of** experiments conducted at our University and at software companies.

(GGM1.3) **Analyse** differences in the level of the GenderMag facets, **for the purpose of** evaluation, **with respect to** their effects on the *ease to create* requirements models, **from the viewpoint of** researchers, **in the context of** experiments conducted at our University and at software companies.

(GGM2) **Analyse** differences in the level of the GenderMag facets, **for the purpose of** evaluation, **with respect to** their effects on the *modification* of requirements models, **from the viewpoint of** researchers, **in the context of** experiments conducted at our University and at software companies.

(GGM2.1) **Analyse** differences in the level of the GenderMag facets, **for the purpose of** evaluation, **with respect to** their effects on the *accuracy to modify* requirements models, **from the viewpoint of** researchers, **in the context of** experiments conducted at our University and at software companies.

(GGM2.2) **Analyse** differences in the level of the GenderMag facets, **for the purpose of** evaluation, **with respect to** their effects on the *speed to modify* requirements models, **from the viewpoint of** researchers, **in the context of** experiments conducted at our University and at software companies.

(GGM2.3) **Analyse** differences in the level of the GenderMag facets, **for the purpose of** evaluation, **with respect to** their effects on the *ease to modify* requirements models, **from the viewpoint of** researchers, **in the context of** experiments conducted at our University and at software companies.

(GGM4) **Analyse** differences in the level of the GenderMag facets, **for the purpose of** evaluation, **with respect to** their effects on the *reviewing* of requirements models, **from the viewpoint of** researchers, **in the context of** experiments conducted at our University and at software companies.

(GGM4.1) **Analyse** differences in the level of the GenderMag facets, **for the purpose of** evaluation, **with respect to** their effects on the *accuracy to review* requirements models, **from the viewpoint of** researchers, **in the context of** experiments conducted at our University and at software companies.

(GGM4.2) **Analyse** differences in the level of the GenderMag facets, **for the purpose of** evaluation, **with respect to** their effects on the *speed to review* requirements models, **from the viewpoint of** researchers, **in the context of** experiments conducted at our University and at software companies.

(GGM4.3) **Analyse** differences in the level of the GenderMag facets, **for the purpose of** evaluation, **with respect to** their effects on the *ease to review* requirements models, **from the viewpoint of** researchers, **in the context of** experiments conducted at our University and at software companies.

8.1.2 Hypotheses

For each one of the goals presented in Subsection 8.1.1, we define the **null** (H_0) and **alternative hypotheses** (H_1). Following the same principle of the goals, all the hypotheses are similar, only changing the *underline and italic* part. However, they are fully specified for documentation purposes and easier reference.

The first set of hypotheses is related with the **requirements languages** ($\boldsymbol{H_{0Lx}}$ and $\boldsymbol{H_{1Lx}}$) themselves, with the objective of comparing the differences between the results achieved when using *i** 1.0, iStar 2.0, ARNE, and ALCO.

$\boldsymbol{H_{0L1}}$ Differences in the requirements languages **do not** influence the *creation* of requirements models.

- $\boldsymbol{H_{0L1.1}}$ Differences in the requirements languages **do not** influence the *accuracy to create* requirements models.
- $\boldsymbol{H_{0L1.2}}$ Differences in the requirements languages **do not** influence the *speed to create* requirements models.
- $\boldsymbol{H_{0L1.3}}$ Differences in the requirements languages **do not** influence the *ease to create* requirements models.

$\boldsymbol{H_{1L1}}$ Differences in the requirements languages influence the *creation* of requirements models.

- $\boldsymbol{H_{1L1.1}}$ Differences in the requirements languages influence the *accuracy to create* requirements models.
- $\boldsymbol{H_{1L1.2}}$ Differences in the requirements languages influence the *speed to create* requirements models.
- $\boldsymbol{H_{1L1.3}}$ Differences in the requirements languages influence the *ease to create* requirements models.

$\boldsymbol{H_{0L2}}$ Differences in the requirements languages **do not** influence the *modification* of requirements models.

- $\boldsymbol{H_{0L2.1}}$ Differences in the requirements languages **do not** influence the *accuracy to modify* requirements models.
- $\boldsymbol{H_{0L2.2}}$ Differences in the requirements languages **do not** influence the *speed to modify* requirements models.
- $\boldsymbol{H_{0L2.3}}$ Differences in the requirements languages **do not** influence the *ease to modify* requirements models.

$\boldsymbol{H_{1L2}}$ Differences in the requirements languages influence the *modification* of requirements models.

$H_{1L2.1}$ Differences in the requirements languages influence the *accuracy to modify* requirements models.

$H_{1L2.2}$ Differences in the requirements languages influence the *speed to modify* requirements models.

$H_{1L2.3}$ Differences in the requirements languages influence the *ease to modify* requirements models.

H_{0L3} Differences in the requirements languages **do not** influence the *understanding* of requirements models.

$H_{0L3.1}$ Differences in the requirements languages **do not** influence the *accuracy to understand* requirements models.

$H_{0L3.2}$ Differences in the requirements languages **do not** influence the *speed to understand* requirements models.

$H_{0L3.3}$ Differences in the requirements languages **do not** influence the *ease to understand* requirements models.

H_{1L3} Differences in the requirements languages influence the *understanding* of requirements models.

$H_{1L3.1}$ Differences in the requirements languages influence the *accuracy to understand* requirements models.

$H_{1L3.2}$ Differences in the requirements languages influence the *speed to understand* requirements models.

$H_{1L3.3}$ Differences in the requirements languages influence the *ease to understand* requirements models.

H_{0L4} Differences in the requirements languages **do not** influence the *reviewing* of requirements models.

$H_{0L4.1}$ Differences in the requirements languages **do not** influence the *accuracy to review* requirements models.

$H_{0L4.2}$ Differences in the requirements languages **do not** influence the *speed to review* requirements models.

$H_{0L4.3}$ Differences in the requirements languages **do not** influence the *ease to review* requirements models.

H_{1L4} Differences in the requirements languages influence the *reviewing* of requirements models.

$H_{1L4.1}$ Differences in the requirements languages influence the *accuracy to review* requirements models.

$H_{1L4.2}$ Differences in the requirements languages influence the *speed to review* requirements models.

$H_{1L4.3}$ Differences in the requirements languages influence the *ease to review* requirements models.

The second set of hypotheses is related with the levels of the GenderMag facets (H_{0GMx} and H_{1GMx}), with the objective of comparing the differences between the personas on each of the 5 (five) problem-solving facets.

H_{0GM1} Differences in the level of the GenderMag facets **do not** influence the *creation* of requirements models.

$H_{0GM1.1}$ Differences in the level of the GenderMag facets **do not** influence the *accuracy to create* requirements models.

$H_{0GM1.2}$ Differences in the level of the GenderMag facets **do not** influence the *speed to create* requirements models.

$H_{0GM1.3}$ Differences in the level of the GenderMag facets **do not** influence the *ease to create* requirements models.

H_{1GM1} Differences in the level of the GenderMag facets influence the *creation* of requirements models.

$H_{1GM1.1}$ Differences in the level of the GenderMag facets influence the *accuracy to create* requirements models.

$H_{1GM1.2}$ Differences in the level of the GenderMag facets influence the *speed to create* requirements models.

$H_{1GM1.3}$ Differences in the level of the GenderMag facets influence the *ease to create* requirements models.

H_{0GM2} Differences in the level of the GenderMag facets **do not** influence the *modification* of requirements models.

$H_{0GM2.1}$ Differences in the level of the GenderMag facets **do not** influence the *accuracy to modify* requirements models.

$H_{0GM2.2}$ Differences in the level of the GenderMag facets **do not** influence the *speed to modify* requirements models.

$H_{0GM2.3}$ Differences in the level of the GenderMag facets **do not** influence the *ease to modify* requirements models.

H_{1GM2} Differences in the level of the GenderMag facets influence the *modification* of requirements models.

$H_{1GM2.1}$ Differences in the level of the GenderMag facets influence the *accuracy to modify* requirements models.

$H_{1GM2.2}$ Differences in the level of the GenderMag facets influence the *speed to modify* requirements models.

$H_{1GM2.3}$ Differences in the level of the GenderMag facets influence the *ease to modify* requirements models.

H_{0GM3} Differences in the level of the GenderMag facets **do not** influence the *understanding* of requirements models.

$H_{0GM3.1}$ Differences in the level of the GenderMag facets **do not** influence the *accuracy to understand* requirements models.

$H_{0GM3.2}$ Differences in the level of the GenderMag facets **do not** influence the *speed to understand* requirements models.

$H_{0GM3.3}$ Differences in the level of the GenderMag facets **do not** influence the *ease to understand* requirements models.

H_{1GM3} Differences in the level of the GenderMag facets influence the *understanding* of requirements models.

$H_{1GM3.1}$ Differences in the level of the GenderMag facets influence the *accuracy to understand* requirements models.

$H_{1GM3.2}$ Differences in the level of the GenderMag facets influence the *speed to understand* requirements models.

$H_{1GM3.3}$ Differences in the level of the GenderMag facets influence the *ease to understand* requirements models.

H_{0GM4} Differences in the level of the GenderMag facets **do not** influence the *reviewing* of requirements models.

$H_{0GM4.1}$ Differences in the level of the GenderMag facets **do not** influence the *accuracy to review* requirements models.

$H_{0GM4.2}$ Differences in the level of the GenderMag facets **do not** influence the *speed to review* requirements models.

$H_{0GM4.3}$ Differences in the level of the GenderMag facets **do not** influence the *ease to review* requirements models.

H_{1GM4} Differences in the level of the GenderMag facets influence the *reviewing* of requirements models.

$H_{1GM4.1}$ Differences in the level of the GenderMag facets influence the *accuracy to review* requirements models.

$H_{1GM4.2}$ Differences in the level of the GenderMag facets influence the *speed to review* requirements models.

$H_{1GM4.3}$ Differences in the level of the GenderMag facets influence the *ease to review* requirements models.

For *requirements languages*, the **independent variable** is the *modelling language*, which may be *i** 1.0, iStar 2.0, ARNE use case template, or ALCO use case template. For *GenderMag*, the variable is the level of the facet – the *persona* – which may be Abby or Tim, on each of the 5 (five) facets (motivation for using software, information processing style, computer self-efficacy, attitude towards risk, and ways of learning new technology).

The **dependent variables** are *accuracy*, *speed*, *ease* (*visual*, *mental*, and *emotional*), and *perceived effort*. The variables and the corresponding metrics were fully described in Subsection 5.1.5.

8.2 Analysis

8.2.1 Hypotheses Testing

We started by applying the *Levene's test* for homogeneity of variance to assess if each group of the independent variable had the same variance. If the Levene statistic is significant at the $p < .05$ level, we reject the null hypothesis that the groups have equal variances. For testing our hypotheses, we used the *Welch's t-test*, as it is robust to deviations from the normal distribution, different sample sizes, and variance in the samples, thus following the recommendations on data analysis for Software Engineering empirical evaluations [121] (which summarises best practices in statistical analysis on other domains). We are using $p < .05$ for the level of significance and thus rejecting the null hypothesis. With more than two groups, Welch's *t*-test does not inform us which groups are different from the others, only that a difference exists. After finding a significant difference, we need to apply a post-hoc test on the factor to examine the differences between the requirements languages. We used the *Games-Howell* post-hoc procedure, which is robust for unequal variances and sample sizes in the groups. We are using $p < .05$ for the level of significance.

In HGM_x, related with the levels of the GenderMag facets, we are also interested in comparing the different levels with the requirements languages, so we used the *Factorial ANOVA test*.

For the sake of brevity, we only present the results concerning RQL1, which serve to illustrate the results for the hypotheses testing. Due to its high number, the remainder of the data can be found in a webpage [213]. In this Section, we only present the results for the hypotheses testing. The discussion on the data can be found in Section 6.4.

RQL1: Does a difference in the requirements languages (i* 1.0, iStar 2.0, ARNE use case template, and ALCO use case template) influence the ability to* create *requirements models?

In Table 8.1 we summarise the Levene's test and the Welch *t*-test results for the creation task, when comparing requirements languages. There was a statistically significant difference in almost all of the variables ($p < .05$), with the *p*-value marked **bold** in the *Sig.* columns of the Table.

Table 8.1: Levene's test and Welch *t*-test: *creation* task, requirements languages.

	Metric	Levene's test Sig. (*p*-value)	Statistic	df1	df2	Welch t-test Sig. (*p*-value)
Accuracy	Precision	.954	23.874	3	90.492	**.000**
	Recall	**.035**	13.391	3	90.326	**.000**
	F-measure	**.012**	32.399	3	87.793	**.000**
Speed	Duration	**.000**	114.955	3	87.636	**.000**
	FirstAct	**.000**	160.872	3	88.168	**.000**
	LastAct	**.000**	88.720	3	87.834	**.000**
	ProcDur	**.000**	18.991	3	91.531	**.000**
Visual ease	FixRel	.084	1133.726	3	89.714	.000
	FixIrrel	**.000**	64.368	3	85.925	**.000**
	AvgDurRelFix	**.000**	367.210	3	91.030	**.000**
	AvgDurIrrelFix	**.000**	100.849	3	91.209	**.000**
	TotSac	.791	114.676	3	91.200	**.000**
	Sac2Key	**.042**	715.014	3	91.634	**.000**
Mental ease	AvgAttention	**.020**	36.195	3	90.232	**.000**
	AvgMentWL	.600	53.750	3	91.612	**.000**
	AvgFam	.384	52.710	3	89.798	**.000**
Emot. ease	AvgSCL	.357	37.517	3	90.468	**.000**
	AvgRMSSD	.739	2.681	3	90.555	.052
	AvgNN50	.893	.315	3	90.949	.814
Perceived effort	Mental demand	**.037**	118.360	3	91.254	**.000**
	Physical demand	.527	.496	3	91.072	.686
	Temporal demand	**.000**	23.379	3	90.490	**.000**
	Effort	.386	53.410	3	91.393	**.000**
	Performance	**.000**	12.665	3	91.502	**.000**
	Frustration	**.000**	32.685	3	92.215	**.000**
	NASA-TLX Score	**.000**	35.708	3	90.782	**.000**

In Table 8.2 we present the Games-Howell post-hoc test concerning *accuracy*, including *precision*, *recall* and *f-measure*. Due to its high number, the remainder of the data can be found in a webpage [213]. There was a statistically significant difference in several of the variables ($p < .05$), with the *p*-value marked **bold** in the *Sig.* column of the Table.

Table 8.2: Games-Howell post-hoc test: *creation* task, requirements languages.

	Req. language (A)	Req. language (B)	Mean (A-B)	Std. Error	Sig.
Precision	*i** 1.0	iStar 2.0	-.085	.044	.221
		ARNE	-.246	.046	**.000**
		ALCO	-.382	.049	**.000**
	iStar 2.0	*i** 1.0	.085	.044	.221
		ARNE	-.161	.045	**.003**
		ALCO	-.297	.048	**.000**
	ARNE	*i** 1.0	.246	.046	**.000**
		iStar 2.0	.161	.045	**.003**
		ALCO	-.136	.050	**.042**
	ALCO	*i** 1.0	.382	.049	**.000**
		iStar 2.0	.297	.048	**.000**
		ARNE	.136	.050	**.042**
Recall	*i** 1.0	iStar 2.0	-.134	.047	**.029**
		ARNE	-.181	.041	**.000**
		ALCO	-.323	.052	**.000**
	iStar 2.0	*i** 1.0	.134	.047	**.029**
		ARNE	-.047	.042	.688
		ALCO	-.189	.053	**.004**
	ARNE	*i** 1.0	.181	.041	**.000**
		iStar 2.0	.047	.042	.688
		ALCO	-.142	.048	**.021**
	ALCO	*i** 1.0	.323	.052	**.000**
		iStar 2.0	.189	.053	**.004**
		ARNE	.142	.048	**.021**
F-measure	*i** 1.0	iStar 2.0	-.091	.033	**.031**
		ARNE	-.256	.038	**.000**
		ALCO	-.399	.046	**.000**
	iStar 2.0	*i** 1.0	.092	.033	**.031**
		ARNE	-.165	.035	**.000**
		ALCO	-.307	.043	**.000**
	ARNE	*i** 1.0	.256	.03816	**.000**
		iStar 2.0	.165	.035	**.000**
		ALCO	-.143	.048	**.020**
	ALCO	*i** 1.0	.399	.04579	**.000**
		iStar 2.0	.307	.043	**.000**
		ARNE	.143	.048	**.020**

8.3 Discussion

8.3.1 Evaluation of Results and Implications

RQL1: Does a difference in the requirements language influence the ability to* create *requirements models?

Assessing accuracy. There was a statistically significant difference among the requirements languages, in terms of accuracy. Participants were able to achieve the highest *precision*, *recall*, and *f-measure* when using ALCO template, followed by ARNE, iStar 2.0, and *i** 1.0. When comparing the two extreme languages, ALCO had a *precision* and *recall* of $\approx 80\%$, while for *i** 1.0 the values were lower than 50%. Our participants had a significantly better accuracy with a textual representation than with a diagrammatic one. In general, participants had little to no prior knowledge on *i**, although some participants had learnt *i** 1.0 in the context of a course. On the other hand, several participants said to have learnt use cases in the context of a course and some of them are still using them in a professional environment. This difference in the knowledge of the requirements languages might have influenced the results. With training, we are confident that participants would be able to improve their performance with *i**. The relationship among the requirements languages, in terms of accuracy (*acc*), can be translated as in Equation 8.1.

$$acc(\text{ALCO}) > acc(\text{ARNE}) > acc(\text{iStar 2.0}) > acc(i^* \text{ 1.0}) \tag{8.1}$$

Assessing speed. There was a statistically significant difference among the requirements languages, in terms of speed. The *duration* of participants using *i** 1.0 was lower than the one of participants using iStar 2.0. However, it was higher than both ARNE and ALCO. In fact, participants using ALCO were faster than any other. For the remaining metrics, the positions of *i** 1.0 and iStar 2.0 are reversed. However, the *processing duration* was lower when using *i** 1.0, meaning the participants do not perform a thorough revision of their work. We argue that this lack of a final analysis jeopardised the results. The relationship among the requirements languages, in terms of the overall speed (*speed*), can be translated as in Equation 8.2.

$$speed(\text{ALCO}) > speed(\text{ARNE}) > speed(i^* \text{ 1.0}) > speed(\text{iStar 2.0}) \tag{8.2}$$

Assessing visual ease. There was a statistically significant difference among the requirements languages, in terms of visual ease. Yet, the values for the several metrics were not consistent, making it not possible to draw a conclusion in terms of the overall visual ease. Participants using ARNE had a greater visual effort, observable through a higher *fixation rate on relevant elements* and *fixation rate on irrelevant elements*. However, the *average duration of relevant fixations* and the *average duration of irrelevant fixations* were lower. On the other hand, participants using *i** 1.0 had a higher *total number of saccades* and *total number of saccades to the key*. These participants were rapidly changing from one model element to the next and thus making a more erratic navigation. Having these

differences into account, we have divided the visual ease into 3 (three) parts, related with number of fixations, duration of fixations, and saccades. The relationship among the requirements languages can be translated as in Equation 8.3, for number of fixations ($nfix$); as in Equation 8.4, for duration of fixations ($dfix$); and as in Equation 8.5, for saccades ($sacc$).

$$nfix(\text{ARNE}) > nfix(i^*\ 1.0) > nfix(\text{ALCO}) > nfix(iStar2.0) \tag{8.3}$$

$$dfix(\text{iStar 2.0}) > dfix(i^*\ 1.0) > dfix(\text{ALCO}) > dfix(\text{ARNE}) \tag{8.4}$$

$$sacc(i^*\ 1.0) > sacc(\text{iStar 2.0}) > sacc(\text{ARNE}) > sacc(\text{ALCO}) \tag{8.5}$$

Assessing mental ease. There was a statistically significant difference among the requirements languages, in terms of mental ease. Participants had a higher *average attention* and *average mental workload* when using i^* 1.0, followed by iStar 2.0, ARNE and ALCO. When comparing the two extreme languages, i^* 1.0 had values of $\approx 80\%$, while ALCO was lower than 40%. Participants were more mentally engaged and attentive to the task they were performing when using i^* 1.0. The relationship among the requirements languages, in terms of mental ease ($mease$), can be translated as in Equation 8.6.

$$mease(\text{ALCO}) > mease(\text{ARNE}) > mease(\text{iStar 2.0}) > mease(i^*\ 1.0) \tag{8.6}$$

Assessing emotional ease. There was no statistically significant difference among the requirements languages in terms of *heart rate variability*. However, there was a difference in terms of *average skin conductive level*. Participants using iStar 2.0 had a higher value for this metric than any other. These participants were more stressed while performing the task. Our interpretation is that they have felt the pressure for achieving a high performance, and an evaluation apprehension. The relationship among the requirements languages, in terms of the average skin conductive level ($avgscl$), can be translated as in Equation 8.7.

$$avgscl(\text{iStar 2.0}) > avgscl(i^*\ 1.0) > avgscl(\text{ALCO}) > avgscl(\text{ARNE}) \tag{8.7}$$

Assessing perceived effort. There was a statistically significant difference among the requirements languages, in terms of perceived effort. These differences is between i^* and the use case templates, and in terms of *mental demand*, *temporal demand* and *effort*. Overall, participants found the task performed on i^* as being more mentally challenging, time consuming, and strenuous. The relationship among the requirements languages, in terms of the perceived effort ($peffort$), can be translated as in Equation 8.8.

$$peffort(i^*\ 1.0) > peffort(\text{iStar 2.0}) > peffort(\text{ARNE}) > peffort(\text{ALCO}) \tag{8.8}$$

RQL2: Does a difference in the requirements language influence the ability to* modify *requirements models?

Assessing accuracy. There was a statistically significant difference among the requirements languages, in terms of accuracy. Participants were able to achieve the highest *precision*, *recall*, and *f-measure* when using ALCO template, followed by ARNE, iStar 2.0, and *i** 1.0. When comparing the two extreme languages, ALCO had a *precision* and *recall* of $\approx$ 80%, while for *i** 1.0 the values were lower than $\approx$ 50%. Our participants had a significantly better accuracy with a textual representation than with a diagrammatic one. In general, participants had little to no prior knowledge on *i**, although some participants had learnt *i** 1.0 in the context of a course. On the other hand, several participants said to have learnt use cases in the context of a course and some of them are still using them in a professional environment. This difference in the knowledge of the requirements languages might have influenced the results. With training, we are confident that participants would be able to improve their performance with *i**. The relationship among the requirements languages, in terms of accuracy (*acc*), can be translated as in Equation 8.9.

$$acc(\text{ALCO}) > acc(\text{ARNE}) > acc(\text{iStar 2.0}) > acc(i^* \text{ 1.0}) \tag{8.9}$$

Assessing speed. There was a statistically significant difference among the requirements languages, in terms of speed. The *duration* of participants using *i** 1.0 was lower than the one of participants using iStar 2.0. However, it was higher than both ARNE and ALCO. In fact, participants using ALCO were faster than any other. For the remaining metrics, the positions of *i** 1.0 and iStar 2.0 are reversed. However, the *processing duration* was lower when using *i** 1.0, meaning the participants do not perform a thorough revision of their work. We argue that this lack of a final analysis jeopardised the results. The relationship among the requirements languages, in terms of the overall speed (*speed*), can be translated as in Equation 8.10.

$$speed(\text{ALCO}) > speed(\text{ARNE}) > speed(i^* \text{ 1.0}) > speed(\text{iStar 2.0}) \tag{8.10}$$

Assessing visual ease. There was a statistically significant difference among the requirements languages, in terms of visual ease. Yet, the values for the several metrics were not consistent, making it not possible to draw a conclusion in terms of the overall visual ease. Participants using ARNE had a greater visual effort, observable through a higher *fixation rate on relevant elements* and *fixation rate on irrelevant elements*. However, the *average duration of relevant fixations* and the *average duration of irrelevant fixations* were lower. On the other hand, participants using *i** 1.0 had a higher *total number of saccades* and *total number of saccades to the key*. These participants were rapidly changing from one model element to the next and thus making a more erratic navigation. Having these differences into account, we have divided the visual ease into 3 (three) parts, related with

number of fixations, duration of fixations, and saccades. The relationship among the requirements languages can be translated as in Equation 8.11, for number of fixations ($nfix$); as in Equation 8.12, for duration of fixations ($dfix$); and as in Equation 8.13, for saccades ($sacc$).

$$nfix(\text{ARNE}) > nfix(i^*\ 1.0) > nfix(\text{ALCO}) > nfix(iStar2.0) \tag{8.11}$$

$$dfix(\text{iStar 2.0}) > dfix(i^*\ 1.0) > dfix(\text{ALCO}) > dfix(\text{ARNE}) \tag{8.12}$$

$$sacc(i^*\ 1.0) > sacc(\text{iStar 2.0}) > sacc(\text{ARNE}) > sacc(\text{ALCO}) \tag{8.13}$$

Assessing mental ease. There was a statistically significant difference among the requirements languages, in terms of mental ease. Participants had a higher *average familiarity* when using the ARNE template, followed by ALCO, i^* 1.0 and iStar 2.0. In general, participants had little to no prior knowledge on i^*, although some participants had learnt i^* 1.0 in the context of a course. On the other hand, several participants said to have learnt use cases in the context of a course and some of them are still using them in a professional environment, which can explain the initial *familiarity* participants had when starting performing the task. The relationship among the requirements languages, in terms of average familiarity ($avgfam$), can be translated as in Equation 8.14.

$$avgfam(\text{ALCO}) > avgfam(\text{ARNE}) > avgfam(i^*\ 1.0) > avgfam(\text{iStar 2.0}) \tag{8.14}$$

Assessing emotional ease. There was a statistically significant difference among the requirements languages, in terms of emotional ease. Participants using iStar 2.0 had a higher *heart rate variability* than any other. Furthermore, these participants also had a higher *average skin conductive level*. They were more stressed and anxious while performing the task. Our interpretation is that these participants felt the pressure for achieving a high performance, as well as an evaluation apprehension. The relationship among the requirements languages, in terms of the emotional ease ($eease$), can be translated as in Equation 8.15.

$$eease(\text{ARNE}) > eease(\text{ALCO}) > eease(i^*\ 1.0) > eease(\text{iStar 2.0}) \tag{8.15}$$

Assessing perceived effort. There was no statistically significant difference between i^* and use cases templates. However, there was a difference between i^* 1.0 and iStar 2.0. The perceived *performance* of participants using i^* 1.0 was lower than the one of participants using iStar 2.0. When using i^* 1.0, participants were not confident on their results. The relationship among the requirements languages, in terms of the perceived performance ($pperf$), can be translated as in Equation 8.16.

$$pperf(\text{ALCO}) > pperf(\text{ARNE}) > pperf(\text{iStar 2.0}) > pperf(i^* \text{ 1.0}) \tag{8.16}$$

RQL3: Does a difference in the requirements language influence the ability to* understand *requirements models?

Assessing accuracy. There was a statistically significant difference among the requirements languages, in terms of accuracy. Participants were able to achieve the highest *precision*, *recall*, and *f-measure* when using ALCO template, followed by ARNE, iStar 2.0, and *i** 1.0. When comparing the two extreme languages, ALCO had a *precision* and *recall* of $\approx 90\%$, while for *i** 1.0 the values were $\approx 70\%$. Our participants had a better accuracy with a textual representation than with a diagrammatic one. In general, participants had little to no prior knowledge on *i**, although some participants had learnt *i** 1.0 in the context of a course. On the other hand, several participants said to have learnt use cases in the context of a course and some of them are still using them in a professional environment. This difference in the knowledge of the requirements languages might have influenced the results. With training, we are confident that participants would be able to improve their performance with *i**. The relationship among the requirements languages, in terms of accuracy (*acc*), can be translated as in Equation 8.17.

$$acc(\text{ALCO}) > acc(\text{ARNE}) > acc(\text{iStar 2.0}) > acc(i^* \text{ 1.0}) \tag{8.17}$$

Assessing speed. There was a statistically significant difference among the requirements languages, in terms of speed. Participants using *i** 1.0 were slower to complete the task than the ones using iStar 2.0, ARNE, and ALCO, having a higher *duration*. They also took more time to perform the *first detection*, as well as to perform the last *detection*. When further comparing the requirements language, participants were undoubtedly faster when understanding a use case specification (less than 5 minutes) than when understanding an *i** model (more than 10 minutes). The relationship among the requirements languages, in terms of the overall speed (*speed*), can be translated as in Equation 8.18.

$$speed(\text{ALCO}) > speed(\text{ARNE}) > speed(\text{iStar 2.0}) > speed(i^* \text{ 1.0}) \tag{8.18}$$

Assessing visual ease. There was a statistically significant difference among the requirements languages, in terms of visual ease. Yet, the values for the several metrics were not consistent, making it not possible to draw a conclusion in terms of the overall visual ease. Participants using ALCO had a greater visual effort, observable through a higher *fixation rate on relevant elements* and *fixation rate on irrelevant elements*. However, the *average duration of relevant fixations* and the *average duration of irrelevant fixations* were lower. On the other hand, participants using *i** 1.0 had a higher *total number of saccades* and *total number of saccades to the key*. These participants were rapidly changing from one model element to the next and thus making a more erratic navigation. Having these

differences into account, we have divided the visual ease into 3 (three) parts, related with number of fixations, duration of fixations, and saccades. The relationship among the requirements languages can be translated as in Equation 8.19, for number of fixations ($nfix$); as in Equation 8.20, for duration of fixations ($dfix$); and as in Equation 8.21, for saccades ($sacc$).

$$nfix(\text{ALCO}) > nfix(\text{ARNE}) > nfix(i^*\ 1.0) > nfix(iStar2.0) \tag{8.19}$$

$$dfix(i^*\ 1.0) > dfix(\text{iStar}\ 2.0) > dfix(\text{ALCO}) > dfix(\text{ARNE}) \tag{8.20}$$

$$sacc(i^*\ 1.0) > sacc(\text{iStar}\ 2.0) > sacc(\text{ARNE}) > sacc(\text{ALCO}) \tag{8.21}$$

Assessing mental ease. There was a statistically significant difference among the requirements languages, in terms of mental ease. Participants had a higher *average attention* when using ARNE, followed by i^* 1.0, ALCO and iStar 2.0. Participants were more mentally engaged and attentive to the task they were performing when using ARNE. Participants also had a higher *average familiarity* when using the ARNE template. In general, participants had little to no prior knowledge on i^*, although some participants had learnt i^* 1.0 in the context of a course. On the other hand, several participants said to have learnt use cases in the context of a course and some of them are still using them in a professional environment, which can explain the initial *familiarity* participants had when starting performing the task. The relationship among the requirements languages, in terms of average familiarity ($avgfam$), can be translated as in Equation 8.22.

$$avgfam(\text{ARNE}) > avgfam(\text{ALCO}) > avgfam(i^*\ 1.0) > avgfam(\text{iStar}\ 2.0) \tag{8.22}$$

Assessing emotional ease. There was no statistically significant difference among the requirements languages in terms of emotional ease. We found no evidence on the impact of requirement language on the emotional ease of participants performing the understanding task on requirements models.

Assessing perceived effort. There was no statistically significant difference among the requirements languages in terms of perceived effort. We found no evidence on the impact of requirement language on the perceived effort of participants performing the understanding task on requirements models.

RQL4: Does a difference in the requirements language influence the ability to* review *requirements models?

Assessing accuracy. There was a statistically significant difference among the requirements languages, in terms of accuracy. Participants were able to achieve the highest

precision, *recall*, and *f-measure* when using ALCO template, followed by ARNE, *i** 1.0 and iStar 2.0. When further comparing the requirements languages, ALCO had a *precision* and *recall* of ≈ 80%, while for *i** the values were lower than 50%. Our participants had a significantly better accuracy with a textual representation than with a diagrammatic one. In general, participants had little to no prior knowledge on *i**, although some participants had learnt *i** 1.0 in the context of a course. On the other hand, several participants said to have learnt use cases in the context of a course and some of them are still using them in a professional environment. This difference in the knowledge of the requirements languages might have influenced the results. With training, we are confident that participants would be able to improve their performance with *i**. The relationship among the requirements languages, in terms of accuracy (*acc*), can be translated as in Equation 8.23.

$$acc(\text{ALCO}) > acc(\text{ARNE}) > acc(i^* \ 1.0) > acc(\text{iStar 2.0}) \tag{8.23}$$

Assessing speed. There was a statistically significant difference among the requirements languages, in terms of speed. The *duration* of participants using *i** 1.0 was lower than the one of participants using iStar 2.0. However, it was higher than both ARNE and ALCO. The processing duration was also lower for *i** 1.0, than for any other, meaning the participants do not performed a thorough revision of their work. We argue that this lack of analysis jeopardised the results. The relationship among the requirements languages, in terms of the overall speed (*speed*), can be translated as in Equation 8.24.

$$speed(\text{ALCO}) > speed(\text{ARNE}) > speed(i^* \ 1.0) > speed(\text{iStar 2.0}) \tag{8.24}$$

Assessing visual ease. There was a statistically significant difference among the requirements languages, in terms of visual ease. Yet, the values for the several metrics were not consistent, making it not possible to draw a conclusion in terms of the overall visual ease. Participants using ARNE had a greater visual effort, observable through a higher *fixation rate on relevant elements* and *fixation rate on irrelevant elements*. However, the *average duration of relevant fixations* and the *average duration of irrelevant fixations* were lower. On the other hand, participants using *i** 1.0 had a higher *total number of saccades* and *total number of saccades to the key*. These participants were rapidly changing from one model element to the next and thus making a more erratic navigation. Having these differences into account, we have divided the visual ease into 3 (three) parts, related with number of fixations, duration of fixations, and saccades. The relationship among the requirements languages can be translated as in Equation 8.25, for number of fixations ($nfix$); as in Equation 8.26, for duration of fixations ($dfix$); and as in Equation 8.27, for saccades (*sacc*).

$$nfix(\text{ARNE}) > nfix(i^* \ 1.0) > nfix(iStar2.0) > nfix(\text{ALCO}) \tag{8.25}$$

$$dfix(\text{iStar 2.0}) > dfix(i^* \ 1.0) > dfix(\text{ALCO}) > dfix(\text{ARNE}) \tag{8.26}$$

$$sacc(i^* \text{ 1.0}) > sacc(\text{iStar 2.0}) > sacc(\text{ALCO}) > sacc(\text{ARNE}) \tag{8.27}$$

Assessing mental ease. There was a statistically significant difference among the requirements languages, in terms of mental ease. Participants had a higher *average attention* and average *mental workload* when using *i** 1.0. On the other hand, participants had a higher familiarity when using the ARNE template. In general, participants had little to no prior knowledge on *i**, although some participants had learnt *i** 1.0 in the context of a course. On the other hand, several participants said to have learnt use cases in the context of a course and some of them are still using them in a professional environment, which can explain the initial *familiarity* participants had when starting performing the task. The relationship among the requirements languages can be translated as in Equation 8.28, for average attention ($avgatt$); as in Equation 8.29 for average mental workload ($avgmentwl$); and as in Equation 8.30 for average familiarity ($avgfam$).

$$avgatt(i^* \text{ 1.0}) > avgatt(\text{iStar 2.0}) > avgatt(\text{ARNE}) > avgatt(\text{ALCO}) \tag{8.28}$$

$$avgmentwl(\text{ARNE}) > avgmentwl(i^* \text{ 1.0}) > avgmentwl(\text{iStar 2.0}) > avgmentwl(\text{ALCO}) \tag{8.29}$$

$$avgfam(\text{ARNE}) > avgfam(i^* \text{ 1.0}) > avgfam(\text{iStar 2.0}) > avgfam(\text{ALCO}) \tag{8.30}$$

Assessing emotional ease. There was a statistically significant difference among the requirements languages, in terms of emotional ease. Participants using iStar 2.0 had a higher *heart rate variability* than any other. Furthermore, these participants also had a higher *average skin conductive level*. They were more stressed and anxious while performing the task. Our interpretation is that these participants felt the pressure for achieving a high performance, as well as an evaluation apprehension. The relationship among the requirements languages, in terms of the emotional ease ($eease$), can be translated as in Equation 8.31.

$$eease(\text{ALCO}) > eease(\text{ARNE}) > eease(i^* \text{ 1.0}) > eease(\text{iStar 2.0}) \tag{8.31}$$

Assessing perceived effort. There was a statistically significant difference among the requirements languages, in terms of perceived effort. This difference is between *i** and the use case templates, and in terms of *mental demand*, *temporal demand* and *effort*. Overall, participants found the task performed on *i** as being more mentally challenging, time consuming, and strenuous. The relationship among the requirements languages, in terms of the perceived effort ($peffort$), can be translated as in Equation 8.32.

$$peffort(i^* \text{ 1.0}) > peffort(\text{iStar 2.0}) > peffort(\text{ARNE}) > peffort(\text{ALCO}) \tag{8.32}$$

RQGM1: Does a difference in the level (Abby and Tim) of each GenderMag facet influence the ability to* create *requirements models?

Assessing accuracy. There was a statistically significant difference among the requirements languages for Abby and Tim, in terms of accuracy. Both Abby and Tim, in all the facets, had a higher *precision*, *recall*, and *f-measure* using the use case templates than those using *i**. This difference was particularly high for Abby in the *motivation*, *information processing*, and *risk* facets. The relationship among the requirements languages, in terms of accuracy of both Abby and Tim ($accAT$), can be translated as in Equation 8.33.

$$accAT(\text{ALCO}) > accAT(\text{ARNE}) > accAT(\text{iStar 2.0}) > accAT(i^* \text{ 1.0}) \tag{8.33}$$

Assessing speed. There was a statistically significant difference among the requirements languages for Abby and Tim, in terms of speed. Both Abby and Tim, in all the facets, had a lower *duration* when using ALCO. Furthermore, the *processing duration* was higher for Tim in the *motivation* facet, when using iStar 2.0, and for Abby in the *self-efficacy* facet when using iStar 2.0 as well. The relationship among the requirements languages, in terms of the overall speed of both Abby and Tim ($speedAT$), can be translated as in Equation 8.34.

$$speedAT(\text{ALCO}) > speedAT(\text{ARNE}) > speedAT(i^* \text{ 1.0}) > speedAT(\text{iStar 2.0}) \tag{8.34}$$

Assessing visual ease. There was a statistically significant difference among the requirements languages for Abby and Tim, in terms of visual ease. Participants characterised as Abby in the *information processing*, *self-efficacy*, and *learning style* facets had a greater visual effort when using *i** 1.0, observable through a higher *fixation rate on irrelevant elements* and *average duration of irrelevant fixations*. Participants identified as Tim, on the other hand, had a greater visual effort when using ARNE. The relationship among the requirements languages, in terms of the overall visual ease of Abby and Tim, can be translated as in Equation 8.35, for Abby ($veaseA$); an as in Equation 8.36, for Tim ($veaseT$).

$$veaseA(\text{ALCO}) > veaseA(\text{ARNE}) > veaseA(iStar2.0) > veaseA(i^* \text{ 1.0}) \tag{8.35}$$

$$veaseT(\text{ALCO}) > veaseT(\text{iStar 2.0}) > veaseT(i^* \text{ 1.0}) > veaseT(\text{ARNE}) \tag{8.36}$$

Assessing mental ease. There was a statistically significant difference among the requirements languages for Abby and Tim, in terms of mental ease. Both Abby and Tim, in all the facets, had a higher *mental workload* when using i* 1.0. However, their average

attention was different. Participants characterised as Abby in the *information processing* facet had a higher *average attention* when using *i** 1.0, but there was no statistically difference in terms of *average attention* for participants characterised as Tim. The relationship among the requirements languages can be translated as in Equation 8.37, for average mental workload of both Abby and Tim ($avgmentwlAT$); and as in Equation 8.38 for average attention of Abby ($avgattA$).

$$\begin{aligned} avgmentwlAT(i^* \text{ 1.0}) > avgmentwlAT(\text{iStar 2.0}) > \\ avgmentwlAT(\text{ARNE}) > avgmentwlAT(\text{ALCO}) \end{aligned} \tag{8.37}$$

$$avgattA(i^* \text{ 1.0}) > avgattA(\text{iStar 2.0}) > avgattA(\text{ARNE}) > avgattA(\text{ALCO}) \tag{8.38}$$

Assessing emotional ease. There was a statistically significant difference among the requirements languages for Abby, in terms of mental ease. Participants characterised as Abby in the *information processing* facet had a higher *heart rate variability*, for *RMSSD*, when using *i** 1.0. There was no statistically significant differences for Tim. The relationship among the requirements languages, in terms of emotional ease of Abby ($eeaseA$), can be translated as in Equation 8.39.

$$eeaseA(\text{ALCO}) > eeaseA(\text{ARNE}) > eeaseA(\text{iStar 2.0}) > eeaseA(i^* \text{ 1.0}) \tag{8.39}$$

Assessing perceived effort. There was a statistically significant difference among the requirements languages for Abby, in terms of perceived effort. Participants characterised as Abby in the *information processing*, *learning style* and *risk* facets had a higher perceived *mental demand* when using *i** 1.0. There was no statistically significant differences for Tim. The relationship among the requirements languages, in terms of the perceived mental demand ($pmentalA$), can be translated as in Equation 8.40.

$$pmentalA(i^* \text{ 1.0}) > pmentalA(\text{iStar 2.0}) > pmentalA(\text{ARNE}) > pmentalA(\text{ALCO}) \tag{8.40}$$

RQGM2: Does a difference in the level (Abby and Tim) of each GenderMag facet influence the ability to* modify *requirements models?

Assessing accuracy. There was a statistically significant difference among the requirements languages for Abby and Tim, in terms of accuracy. Both Abby and Tim, in all the facets, had a higher *precision*, *recall*, and *f-measure* using the use case templates than those using *i**. This difference was particularly high for Abby in the *motivation*, *self-efficacy*, *information processing*, and *risk* facets. The relationship among the requirements languages, in terms of accuracy of both Abby and Tim ($accAT$), can be translated as in Equation 8.41.

$$accAT(\text{ALCO}) > accAT(\text{ARNE}) > accAT(\text{iStar 2.0}) > accAT(i^* \text{ 1.0}) \tag{8.41}$$

Assessing speed. There was a statistically significant difference among the requirements languages for Abby and Tim, in terms of speed. Both Abby and Tim, in all the facets, had a lower *duration* when using ALCO. Furthermore, the *processing duration* was higher for Tim in the *motivation* facet, when using iStar 2.0, and for Abby in the *self-efficacy* facet when using iStar 2.0 as well. The relationship among the requirements languages, in terms of the overall speed of both Abby and Tim ($speedAT$), can be translated as in Equation 8.42.

$$speedAT(\text{ALCO}) > speedAT(\text{ARNE}) > speedAT(i^*\ 1.0) > speedAT(\text{iStar 2.0}) \tag{8.42}$$

Assessing visual ease. There was a statistically significant difference among the requirements languages for Abby, in terms of visual ease. Participants characterised as Abby in the *information processing*, *learning style* and *risk* facets had a greater visual effort when using i^* 1.0, observable through a higher *fixation rate on irrelevant elements* and *average duration of irrelevant fixations*. There was no statistically significant differences for Tim. The relationship among the requirements languages, in terms of the visual ease of Abby, can be translated as in Equation 8.43.

$$veaseA(\text{ALCO}) > veaseA(\text{ARNE}) > veaseA(iStar2.0) > veaseA(i^*\ 1.0) \tag{8.43}$$

Assessing mental ease. There was a statistically significant difference among the requirements languages for Abby and Tim, in terms of mental ease. Both Abby and Tim, in the *risk* facet, had a higher *mental workload* when using i^* 1.0. However, their average attention was different. Participants characterised as Abby in the *information processing* and *self-efficacy* facets had a higher *average attention* when using i^* 1.0, but there was no statistically difference in terms of *average attention* for participants characterised as Tim. The relationship among the requirements languages can be translated as in Equation 8.44, for average mental workload of both Abby and Tim ($avgmentwlAT$); and as in Equation 8.45 for average attention of Abby ($avgattA$).

$$\begin{aligned} avgmentwlAT(i^*\ 1.0) > avgmentwlAT(\text{iStar 2.0}) > \\ avgmentwlAT(\text{ARNE}) > avgmentwlAT(\text{ALCO}) \end{aligned} \tag{8.44}$$

$$avgattA(i^*\ 1.0) > avgattA(\text{iStar 2.0}) > avgattA(\text{ARNE}) > avgattA(\text{ALCO}) \tag{8.45}$$

Assessing emotional ease. There was a statistically significant difference among the requirements languages for Abby, in terms of mental ease. Participants characterised as Abby in the *risk* facet had a higher *heart rate variability*, for *RMSSD*, when using i^* 1.0. There was no statistically significant differences for Tim. The relationship among the requirements languages, in terms of emotional ease of Abby ($eeaseA$), can be translated as in Equation 8.46.

$$eeaseA(\text{ALCO}) > eeaseA(\text{ARNE}) > eeaseA(\text{iStar 2.0}) > eeaseA(i^*\ 1.0) \tag{8.46}$$

Assessing perceived effort. There was a statistically significant difference among the requirements languages for Abby, in terms of perceived effort. Participants characterised as Abby in the *information processing*, *self-efficacy* and *risk* facets had a higher perceived *mental demand* and *temporal demand* when using i^* 1.0. There was no statistically significant differences for Tim. The relationship among the requirements languages, in terms of the perceived effort ($pef fortA$), can be translated as in Equation 8.47.

$$pef fortA(i^*\ 1.0) > pef fortA(\text{iStar 2.0}) > pef fortA(\text{ARNE}) > pef fortA(\text{ALCO}) \tag{8.47}$$

RQGM3: Does a difference in the level (Abby and Tim) of each GenderMag facet influence the ability to* understand *requirements models?

Assessing accuracy. There was a statistically significant difference among the requirements languages for Abby and Tim, in terms of accuracy. Both Abby and Tim, in all the facets, had a higher *precision* and *recall*, using the use case templates than those using i^*. This difference was particularly high for Abby in the *information processing* and *risk* facets. The relationship among the requirements languages, in terms of accuracy of both Abby and Tim ($accAT$), can be translated as in Equation 8.48.

$$accAT(\text{ALCO}) > accAT(\text{ARNE}) > accAT(\text{iStar 2.0}) > accAT(i^*\ 1.0) \tag{8.48}$$

Assessing speed. There was a statistically significant difference among the requirements languages for Abby and Tim, in terms of speed. Participants characterised as Abby in the *motivation*, *learning style* and *information processing* facets had a lower *duration* when using ALCO. Participants identified as Tim in *information processing* and *risk* facets also had a lower duration when using ALCO. Furthermore, the *processing duration* was higher for Tim in the *information processing* facet, when using i^* 1.0, and for Abby in the *self-efficacy* facet when using i^* 1.0 as well. The relationship among the requirements languages, in terms of the overall speed of both Abby and Tim ($speedAT$), can be translated as in Equation 8.49.

$$speedAT(\text{ALCO}) > speedAT(\text{ARNE}) > speedAT(i^*\ 1.0) > speedAT(\text{iStar 2.0}) \tag{8.49}$$

Assessing visual ease. There was a statistically significant difference among the requirements languages for Abby, in terms of visual ease. Participants characterised as Abby in the *information processing*, *self-efficacy*, and *learning style* facets had a greater visual effort when using i^* 1.0, observable through a higher *fixation rate on irrelevant*

elements and *average duration of irrelevant fixations*. There was no statistically significant differences for Tim. The relationship among the requirements languages, in terms of the overall visual ease of Abby (*veaseA*), can be translated as in Equation 8.50.

$$veaseA(\text{ALCO}) > veaseA(\text{ARNE}) > veaseA(iStar2.0) > veaseA(i^{*}\ 1.0) \tag{8.50}$$

Assessing mental ease. There was a statistically significant difference among the requirements languages for Abby, in terms of mental ease. Participants characterised as Abby, independently on the facet, had a higher *average mental workload* when using *i** 1.0. Furthermore, Abby in the *information processing* and *self-efficacy* facets also had a higher *average attention* when using *i** 1.0. There was no statistically significant differences for Tim. The relationship among the requirements languages, in terms of the mental ease of Abby (*measeA*), can be translated as in Equation 8.51.

$$measeA(\text{ALCO}) > measeA(\text{ARNE}) > measeA(\text{iStar 2.0}) > measeA(i^{*}\ 1.0) \tag{8.51}$$

Assessing emotional ease. There was a statistically significant difference among the requirements languages for Abby, in terms of mental ease. Participants characterised as Abby in the *information processing* and *risk* facets had a higher *heart rate variability*, for *RMSSD*, when using *i** 1.0. There was no statistically significant differences for Tim. The relationship among the requirements languages, in terms of emotional ease of Abby (*eeaseA*), can be translated as in Equation 8.52.

$$eeaseA(\text{ALCO}) > eeaseA(\text{ARNE}) > eeaseA(\text{iStar 2.0}) > eeaseA(i^{*}\ 1.0) \tag{8.52}$$

Assessing perceived effort. There was a statistically significant difference among the requirements languages for Abby, in terms of perceived effort. Participants characterised as Abby in the *self-efficacy* and *risk* facets facets had a higher perceived *mental demand*, *temporal demand* and *frustration* when using *i** 1.0. There was no statistically significant differences for Tim. The relationship among the requirements languages, in terms of the perceived mental demand (*pmentalA*), can be translated as in Equation 8.53.

$$pef fortA(i^{*}\ 1.0) > pef fortA(\text{iStar 2.0}) > pef fortA(\text{ARNE}) > pef fortA(\text{ALCO}) \tag{8.53}$$

RQGM4: Does a difference in the level (Abby and Tim) of each GenderMag facet influence the ability to* review *requirements models?

Assessing accuracy. There was a statistically significant difference among the requirements languages for Abby and Tim, in terms of accuracy. Both Abby and Tim, in all the facets, had a higher *precision* and *recall* using the use case templates than those using

*i**. This difference was particularly high for Abby in the *self-efficacy* and *risk* facets. The relationship among the requirements languages, in terms of accuracy of both Abby and Tim (*accAT*), can be translated as in Equation 8.54.

$$accAT(\text{ALCO}) > accAT(\text{ARNE}) > accAT(\text{iStar 2.0}) > accAT(i^*\ 1.0) \tag{8.54}$$

Assessing speed. There was a statistically significant difference among the requirements languages for Abby and Tim, in terms of speed. Participants characterised as Abby in the *self-efficacy* facet had a lower *duration* when using ALCO. Participants identified as Tim in *information processing* facet also had a lower duration when using ALCO. Furthermore, the *processing duration* was higher for Tim in the *information processing* facet, when using *i** 1.0, and for Abby in the *self-efficacy* facet when using *i** 1.0 as well. The relationship among the requirements languages, in terms of the overall speed of both Abby and Tim (*speedAT*), can be translated as in Equation 8.55.

$$speedAT(\text{ALCO}) > speedAT(\text{ARNE}) > speedAT(i^*\ 1.0) > speedAT(\text{iStar 2.0}) \tag{8.55}$$

Assessing visual ease. There was a statistically significant difference among the requirements languages for Abby, in terms of visual ease. Participants characterised as Abby in the *information processing* facet had a greater visual effort when using *i** 1.0, observable through a higher *fixation rate on irrelevant elements* and *average duration of irrelevant fixations*. There was no statistically significant differences for Tim. The relationship among the requirements languages, in terms of the visual ease of Abby, can be translated as in Equation 8.56.

$$veaseA(\text{ALCO}) > veaseA(\text{ARNE}) > veaseA(iStar2.0) > veaseA(i^*\ 1.0) \tag{8.56}$$

Assessing mental ease. There was a statistically significant difference among the requirements languages for Abby, in terms of mental ease. Participants characterised as Abby in *information processing* and *self-efficacy* facet had a higher *average mental workload* and *average attention* when using *i** 1.0. There was no statistically significant differences for Tim. The relationship among the requirements languages, in terms of the mental ease of Abby (*measeA*), can be translated as in Equation 8.57.

$$measeA(\text{ALCO}) > measeA(\text{ARNE}) > measeA(\text{iStar 2.0}) > measeA(i^*\ 1.0) \tag{8.57}$$

Assessing emotional ease. There was a statistically significant difference among the requirements languages for Abby, in terms of mental ease. Participants characterised as Abby in the *information processing* and *risk* facets had a higher *heart rate variability*, for *RMSSD*, when using *i** 1.0. There was no statistically significant differences for Tim.

The relationship among the requirements languages, in terms of emotional ease of Abby (*eeaseA*), can be translated as in Equation 8.58.

$$eeaseA(\text{ALCO}) > eeaseA(\text{ARNE}) > eeaseA(\text{iStar 2.0}) > eeaseA(i^* \text{ 1.0}) \tag{8.58}$$

Assessing perceived effort. There was a statistically significant difference among the requirements languages for Abby, in terms of perceived effort. Participants characterised as Abby in the *self-efficacy*, *information processing*, and *risk* facets facets had a higher perceived *mental demand*, and *frustration* when using *i** 1.0. There was no statistically significant differences for Tim. The relationship among the requirements languages, in terms of the perceived mental demand (*pmentalA*), can be translated as in Equation 8.59.

$$pef fortA(i^* \text{ 1.0}) > pef fortA(\text{iStar 2.0}) > pef fortA(\text{ARNE}) > pef fortA(\text{ALCO}) \tag{8.59}$$

8.3.2 Inferences

Textual representations of requirements outperformed diagrammatic ones. For the majority of the metrics, participants were able to achieve a better performance and lower effort when using ALCO, followed by ARNE, iStar 2.0, and *i** 1.0. Our interpretation is that, although natural language can be potentially vague and prone to omissions and ambiguity, the participants are familiar with it, being easier for them to understand it and perform the tasks.

Tacit knowledge may hinder performance of the understanding task, in textual representation of requirements. When analysing the replies of our participants, they only refer to their own experience when using ARNE or ALCO templates. Our interpretation is that, with *i**, participants had to focus on the meaning of the concrete syntax, and their correspondence with the questions being asked. As such, the participants were not assuming conditions about the system. With use cases specifications, on the other hand, the usage of natural language may have cause the participants to be less attentive to the several element of the templates. This possibly made them reply with their knowledge, and not with what was described in the use case specification provided. However, and even with this limitation, participant were still able to achieve better results with use cases than with *i**.

Textual representations of requirements are better suited for Abby. Participants characterised as Abby, independently on the facet, were able to have a better performance and lower effort when using ALCO or ARNE templates. In fact, their best performance was when using ALCO, followed by ARNE, iStar 2.0 and *i** 1.0. Our interpretation is that Abby is more comfortable with things she is more familiar with, in this case, the natural language presented in the use case specifications. However, we argue that, with training, Abby would be able to achieve a higher performance with *i**.

8.4 Summary

In this Chapter, we present a comparison of *i** 1.0, iStar 2.0, ARNE use case template, and ARCO use case template. This comparison is based on the data from the quasi-experiments reported in Chapters 6 and 7. We performed a family of quasi-experiments to analyse the impact of different requirements languages, as well as different levels in each of the five GenderMag facets, when creating, modifying, understanding and reviewing requriements models. We measured the accuracy, speed, ease (visual, mental, and emotional), and perceived effort of a total of 660 participants. We used metrics of task success, time, and effort, collected with eye-tracking, EEG and EDA sensors, and participants' feedback through a NASA-TLX questionnaire.

We found that our participants were able to achieve better results with ALCO, followed by ARNE, iStar 2.0, and i* 1.0. However, tacit knowledge had a great impact on the overall performance, specially in the understanding task of both use case templates. Furthermore, there are several differences in the individual characteristics (the GenderMag facets) of the participants that had an influence on their performance and effort.

Chapter 9

Related Work

In this Chapter we present a review of the state of the art in the fields and topics in which this dissertation takes place. Work related to our research can be categorized into 3 (three) major areas: *(i)* studies evaluating quality through the analysis of the requirements' models themselves; *(ii)* research on the exploitation of human factors and the usage of biometrics to evaluate quality of software artefacts; and *(iii)* investigation on gender differences in how people solve software-related problems. We conclude the Chapter by discussing the main differences between the described research and this dissertation.

9.1 Quality Evaluation by Analysing Requirements Models

Quality assessment of conceptual models has been studied several times over the years, and different frameworks to perform that evaluation have been proposed (see Lindland et al. [134] and Krogstie et al. [128]). Indeed, due to the high number of different proposals, none of them widely accepted in practice, Moody [144] felt that there was a need to have a common evaluation framework, since he considers the proliferation of quality frameworks as being counterproductive. Mayerhofer [140] states that methods for ensuring the quality of models can be divided into two fields: static analysis of models, and dynamic analysis of models. According to her, static analysis methods verify the correctness of models by assessing their static properties, whereas dynamic analysis methods verify the quality of models by executing them. Guizzardi and Sales [88] state that an approach to conceptual modelling requires tools for modellers to gain confidence on the quality of the models they produce, and to be able to develop high-quality models, a modeller must have the support of expressive engineering tools. They proposed a tool that is able to automatically identify anti-patterns in user's OntoUML models, provide visualization for their consequences, and generate corrections to these models by the inclusion of OCL

constraints.

Horkoff and Yu [96] evaluate 7 (seven) goal satisfaction analysis procedures using available tools that implement those procedures. The results help to understand the ways in which procedural design choices affect analysis results, and how differences in analysis results could lead to different recommendations over alternatives in the model. Ramos et al. [176] claim that early identification of syntactical problems (e.g., large and unclear descriptions, duplicated information) and the removal of their causes, can improve the quality of use case models. They describe the AIRDoc approach, which aims to facilitate the identification of potential problems in requirements documents using refactoring and patterns. To evaluate use case models, the AIRDoc process uses the GQM approach.

According to Monperrus et al. [143], metrics are a practical approach to evaluate properties of domain-specific models, but it is costly to develop measurement software for each one of them. They present a model-driven and generative approach to measuring models that is domain-independent. Several studies have been carried out regarding the quality evaluation of requirements models, by using metrics. In this field, some of the most studied quality attributes are understandability and comprehensibility, efficiency, correctness, defect rate, completeness and consistency, confinement and changeability [44, 142]. However, the majority of the studies are related with the quality evaluation of UML models [142]. Eichelberger [56, 57] proposes a set of aesthetic criteria for UML class diagrams and discusses the relationship between those criteria and design aspects of object-oriented software. Furthermore, he presents an implementation of a graph drawing framework which produces UML class diagrams according to set of aesthetic criteria. Berenbach and Borotto [14] describes a CMMI (Capability Maturity Model Integration) compliant approach to measurement and analysis during a model-driven requirements development process. He proposes collecting a set of metrics from UML models in order to evaluate their completeness. The metrics were successfully used on several Siemens projects. Lange et al. [131] developed a tool for collecting metrics about the fulfilment of task on UML models. Furthermore, they propose a visualisation for those metrics.

Regarding the evaluation of other requirements models, Matulevičius and Heymans [139] evaluate how KAOS and its tool, Objectiver, help the modelling activity, offering recommendations for modellers, language designers and tool developers. Espada et al. [62] proposed and validated a metrics suite for evaluating completeness and complexity of KAOS goal models, specified using OCL and incorporated in a KAOS modelling tool. The metrics suite was evaluated with several real world case studies. Furthermore, Franch and Grau [67] proposed a framework for defining metrics in *i** models, to analyse the quality of individual models, and to compare alternative models over certain properties. This framework uses a catalogue of patterns for defining metrics, and OCL to formulate these metrics. In a follow up work, Franch proposed a generic method to better guide the analyst throughout the metrics definition process, over *i** models [66]. The method is applied to evaluate business process performance. In previous a work, we defined, implemented, and validated complexity and completeness metrics for *i** 1.0 models [81,

82]. The metrics were defined using OCL and incorporated in an *i** 1.0 modelling tool.

9.2 Quality Evaluation by Exploring Human Factors

The exploitation of human factors is regarded as a relevant topic in Software Engineering [125]. In particular, the RE community is concerned with bridging the perceived gap between sophisticated requirements engineering approaches and the stakeholders with whom requirements engineers need to interact with. The most common requirements language remains to be natural language [152], but it often leads to ambiguous requirements specifications. On the other hand, specialised RE frameworks are poorly understood by relevant stakeholders. As such, devising ways of making these requirements languages more accessible is perceived as very important. Moody [145] and Caire et al. [32] propose approaches to help improving the understandability of requirements models, by improving the concrete syntax of those models through the definition of a set of principles, for designing cognitively effective visual notations. In that sense, it is important to analyse not only the models as an isolated entity, but also investigate the impact of the models on different activities performed by stakeholders, when interacting with those models.

Several studies use controlled experiments for their research, and evaluate user performance in carrying out software tasks by using paper questionnaires or online surveys. Störrle [214] has studied the impact of the usage of good *versus* bad diagram layouts on model comprehension tasks when using UML, namely use case, class, and activity diagrams. He reported on three controlled experiments with 77 participants, and found a significant difference between the layouts, where a good layout produced better results than a bad layout. Later on, he expanded his initial experiment by varying both diagram types and expertise of the population studied [215] and by analysing different diagram sizes [216]. He found that a good layout is particularly helpful for large diagrams. Furthermore, experts performed better than novices, and the benefit of a good layout was larger for novices than experts. Sharif and Maletic [199] studied how stereotype based layouts impact the understandability of UML class diagrams by using online questionnaires. Participants were given one task addressing UML syntax and another addressing software design. The results show a significant improvement in performance when using a multi-cluster layout.

Another way of investigating the impact of the software artefacts' quality on different activities performed by stakeholders when interacting with those artefacts is through biometrics. Most of the studies in Software Engineering using biometric sensors have focused particularly on **eye-tracking technology** and have investigated how developers understand source code, not requirements models. Crosby and Stelovsky [47] used the eye-tracking technology to study the differences in program comprehension and source code reading navigation strategies between experienced and less experienced software developers in the Pascal programming language. They also compared the differences in reading an algorithm written in Pascal and simple text. The results showed that participants

reading Pascal had a higher number of fixations in most areas of the algorithm, as well as spent more time viewing relevant areas, than those using simple text. Moreover, participants with lower experience devoted significantly more attention to comments than did those with higher experience. Uwano et al. [233] and Bednarik and Tukiainen [11, 12] performed similar studies to investigate different strategies of novices and experts in program comprehension and debugging tasks. These studies found repetitive patterns of visual attention that were associated with less experience in programming and lower performance. Sharif et al. [200] partially replicated the study performed by Uwano et al., and analysed the effectiveness and efficiency of finding defects with respect to eye gaze. The results indicate that scanning significantly correlates with defect detection time, as well as visual effort on relevant defects. Finally, Sharif and Maletic [196] used eye-tracking for small program comprehension tasks to investigate the effect of naming conventions (*camelCase* and *under_score*) on the success of solving the program comprehension tasks. The results indicate no difference, in terms of accuracy, between the two styles. However, participants recognised identifiers in the under_score style more quickly. Over the years, several other studies using eye-tracking devices have been performed to evaluate various programming languages (such as C++, Python and Java), and focusing on topics ranging from code reading strategies to naming conventions (see, for example, [28, 29, 120, 237]).

More recently, eye-tracking has been used on some occasions to assess the effort involved in the understanding of software models [193]. Yusuf et al. [250] used eye-tracking to compare the visual effort involved in answering questions about UML class diagrams containing the same information, but designed following 3 different layout strategies: multiple-cluster (classes with related functionality are in clusters); three-cluster (positions classes based on their stereotype role) and orthogonal layout (minimises edge crossings and bending). They concluded that *multiple-cluster* outperformed *three-cluster* and *orthogonal* layouts, as participants had to make, on average, a smaller number of fixations on the diagram. This study was later extended without eye-tracking, confirming the previous results [198]. In the same line of investigation, Sharif et al. [195, 197] studied the effect of different layouts for design pattern roles identification in UML class diagrams. They concluded that layout plays a significant role in the comprehension of these diagrams. In particular, with the multi-cluster layout, there was a significant improvement in accuracy, time, and visual effort. Lastly, Smet et al. [209] showed that although the presence of a visitor pattern and its layout had no significant impact on the comprehension of UML class diagrams, it did have a significant impact in modification tasks.

A common feature in all these studies is the concern with the importance of some aspect of a UML diagram layout, being that a layout heuristic, or the explicit usage of a particular design pattern. Other studies with eye-tracking focused on BPMN [168], ER diagrams [31], and TROPOS diagrams [192]. In the former, Petrusel and Mendling evaluated the significance of relevant regions in BPMN models, by performing an experiment with expert process modellers. The authors concluded that there is a correlation between the relevant region to where the participant is looking at and the answer given

to a model comprehension question. With ER diagrams, Cagiltay et al. studied the defect detection process of participants using these diagrams, and propose two metrics to better understand the software engineers' reasoning process. The results indicate that detecting missing information in ER diagrams is harder than detecting other types of defects, as well as defining and identifying missing information is harder than recognising it. In the latter, Sharafi et al. contrasted the effectiveness of a textual language and the TROPOS diagrammatic requirements language for requirements comprehension purposes and the textual language turned out to be more effective.

Only very few studies investigated the use of other biometric sensors rather than eye-trackers. Using **brain activity measures**, Ikutani and Uwano [105] used near-infrared spectroscopy to investigate the difference in brain activity for various types of program comprehension tasks. They observed significant differences in brain activity at a task that requires memorising variables to understand a code snippet. On the other hand, no significant differences between different levels of mental arithmetic tasks were observed. Siegmund et al. [204] examined the active brain regions during small code comprehension tasks using functional magnetic resonance imaging (fMRI) technology. The results of their study showed that brain regions related to working memory, attention and language processing are active during program comprehension. Huang et al. [100] used functional near-infrared spectroscopy (fNIRS) and fMRI to study the neural representations of 76 participants while performing tasks on several data structures (list, array, tree) and mental rotations. They concluded that data structures and spatial operations use the same focal regions of the brain but to different degrees. In addition, difficult data structure problems have a higher cognitive load than problems related with pure spatial reasoning. Finally, Parnin [166] used electromyography to measure developers' sub-vocal utterances and found that these utterances might be used to measure programming task difficulty.

One of the early approaches that mentioned **multiple biometric sensors** was Ginger2, an environment for computer-aided empirical software engineering proposed by Torii et al. [228]. They included an eye-tracker and a skin resistance level sensor into an environment that was built to continuously collect data from software developers participating in empirical studies (see, for example, [50, 231]). More recently, Fritz et al. [68] and Störrle et al. [217] propose approaches to classify the difficulty of code or models comprehension, respectively, by using a combination of eye-tracking and EEG activities. Finally, Müller and Fritz [147] used eye-tracking, EDA and EEG to investigate developers' emotions in the context of software change tasks. The results show that the wide range of emotions experienced by developers is correlated with their perceived progress on the change tasks. Later on, the authors used both EDA and EEG to predict code quality online, and automatically identify code quality concerns [148]. They concluded that biometrics are, in fact, able to predict quality concerns of parts of the code, while a developer is working on it.

9.3 Gender Differences in Solving Software-Related Problems

Gender differences in problem solving activities have been investigated in different domains. Byrnes et al. [30] conducted a meta-analysis of 150 studies in which the risk-taking tendencies of male and female participants were compared. The several results indicated a greater tendency for risk taking in male participants. Pajares and Miller [164] investigated the impact of self-efficacy on mathematical problem-solving success. They concluded that men had higher performance, self-efficacy, and self-concept and lower anxiety. However, these differences were due largely to the influence of self-efficacy, since gender had a direct effect only on self-efficacy and prior experience.

In particular, the analysis of gender gap in STEAM (**S**cience, **T**echnology, **E**ngineering, **a**nd **M**athematics) contexts has been increasingly investigated [40]. The profile of a computer scientist seems still to be stereotyped, and women show less interest in computer science and less likelihood to consider it as a possible future career [238]. Furthermore, it is common, in software systems, have features that are inadvertently designed to be more supportive of problem-solving processes typically followed by males than by females [86, 219]. In order to attract more females to computer science, educators have focused on offering various coding experiences specially for girls [119]. Çakir et al. [33] conducted a game-design workshop for girls. At the end of the workshop, the girls had better attitudes towards computer science, higher confidence and self-reported competence with computers. Conversely, an earlier study by Robertson [181], investigated the influence of a game-development project on students' attitudes and the results revealed that the level of enjoyment of the project was higher in boys than in girls. In the end, the project did not increase the possibility of them studying computer science in the future. Bruckman et al. [24] analysed 475 children in a computer-supported collaborative learning (CSCL) and using the MOOSE programming language [25]. The results show that girls spend significantly more time communicating with others in the CSCL environment than boys. When analysing the level of programming achievement, gender has not affected programming performance. In fact, performance was correlated with prior programming experience and time spent on task. However, boys are more likely than girls to have prior programming experience. Finally, Papavlasopoulou et al. [165] designed and evaluate a coding workshop for children. They used an eye-tracker and qualitative data from children's interviews to examine differences between boys and girls in coding activities using Scratch. There was no significant difference between boys and girls gaze and learning gain during the coding activity. However, answer to the interviews showed differences in the strategies and implemented practices during coding.

With adults, Beckwith et al. [9] studied gender differences in the context of debugging spreadsheets. The results indicant a significant gender differences in terms of self-efficacy and feature acceptance, with females exhibiting lower self-efficacy and lower feature acceptance. Moreover, the results also show that these differences can significantly reduce females' effectiveness. Later on, the authors compared males and females in a debugging

setting, and investigated how tinkering behaviour impacts several measures of their debugging success [10]. The authors concluded that the factors of tinkering, reflection, and self-efficacy, can combine in multiple ways to impact debugging effectiveness differently for males than for females. Torkzadeh and Koufteros [229] used a 30-item computer self-efficacy scale to examine the influence of computer training on computer self-efficacy. They collect data from 224 undergraduates at the beginning and at the end of an introductory computer course, and evaluated the impact on female and male participants. The results suggest that training significantly improved the computer self-efficacy in both males and females. Fisher et al. [64] conducted a study to compare male and female participants' performance on program comprehension tasks. They found a correlation suggestion that programmers use equivalently risky strategies for program comprehension and spatial cognition. As such, there is evidence that similar cognitive skills are used for both tasks, and that the similarities are a consequence of gender-based differences in risk-taking behaviour. Gramß et al. [85] examined the relationship between gender and performance in a software engineering course. The results indicated that females benefit from a more practical learning method.

Some studies analysed gender differences in adults by using biometrics. Sharafi et al. [191] used eye-tracking devices to study the differences in reading strategies of female and male participants when working on program comprehension tasks. The results showed that was no difference in terms of gender and precision. However, female participants focused more on incorrect answers than male participants, but this has not affected the total duration of the task. Hou et al. [99] examined the eye-movements and gaze paths of 13 males and 12 female computer science students to understand the gender impact on debugging different source code structure, in C. They reported that female students focused more on the requirements of the program before tracing them into the main parts of the program. On the other hand, male students focused more on the change in output and the logic of the loop in the program. Obaidellah and Haek [158] conducted an eye-tracking study 21 female and 30 male computer science undergraduate university students, with the goal of examining their cognitive processes in pseudocode comprehension. The tasks required students to rearrange randomised pseudocode statements in a correct order. The results indicated that the speed of analysing the problems were faster among male students, although female students fixated longer in understanding the problem requirements. In addition, females more commonly fixated on indicative verbs, while males fixated more on operational statements.

9.4 Discussion

In Table 9.1 we present a summary of the related work. In the first column we show the authors of the study, with the corresponding references. In the second column, we inform whether the object of study was a model or code (i.e., a particular programming language or a programming style). The third column has the goal of the study or what

task the authors were evaluating. In the fourth column, we present the types of metrics or biometrics that were collected. Finally, for the studies that involved experiments, we list, in the last column, the number of participants and whether they were children, university students, or practitioners.

Although some work has been performed regarding the quality evaluation of requirements models (mainly UML class diagrams), the vast majority of research using biometrics focused on source code. Furthermore, the combination of metrics about the models themselves, different types of biometrics (eye-tracking, electroencephalography and electrodermal activity), success rate and time of the performed tasks, the stakeholders perceptions (subjective opinion) on their success and effort, and the impact of stakeholders problem-solving facets have not yet been explored, which we have done in this dissertation. A detailed discussion on the advantages of using multiple techniques was previously presented in Subsection 2.4.8. Moreover, when we analyse the type of participants on the studies that explore human factors, we note that, with few exceptions (see [148, 215]), those participants were Computer Science students. In our research, we have performed several quasi-experiments in various contexts, with different types of participants, including field studies with practitioners.

Table 9.1: Summary of the related work.

Study	Models/Code	Goal/Evaluation	Spec. language/ (Bio)metrics	Participants
Horkoff and Yu [96]	GRL, *i**, NFR, Tropos	design choices	–	–
Ramos et al. [176]	use cases	quality	natural lang.	–
Monperrus et al. [143]	DS models	quality	proto-textual	–
Eichelberger [56, 57]	UML	aesthetics	natural lang.	–
Berenbach and Borotto [14]	UML	completeness	natural lang.	–
Lange et al. [131]	UML	aesthetics	natural lang.	–
Matulevičius and Heymans [139]	KAOS	complexity	natural lang.	–
Espada et al. [62]	KAOS	complexity, completeness	OCL	–
Franch [66, 67]	*i**	metrics definition	OCL	–
Gralha et al. [81, 82]	*i**	complexity, completeness	OCL	–
Moody [145]	req. models	understandability	–	–
Caire et al. [32]	*i**	understandability	–	65 students
Störrle [214]	UML	understandability	success	77 students
Störrle [215]	UML	understandability	success	78 students
Störrle [216]	UML	understandability	success	78 students
Sharif and Maletic [199]	UML	comprehension	eye-tracker	28 students
Crosby and Stelovsky [47]	Pascal	comprehension, review	eye-tracker	19 students

continue on next page...

Table 9.1: *...continued from previous page*

Study	Models/Code	Goal/Evaluation	Spec. language/ (Bio)metrics	Participants
Uwano et al. [233]	C	comprehension, review	eye-tracker, time	15 students
Bednarik and Tukiainen [11, 12]	Java	comprehension	eye-tracker	18 students
Sharif et al. [200]	C++	comprehension, review	eye-tracker, time	15 students
Sharif and Maletic [196]	source code	reading, comprehension	eye-tracker, time, success	15 students
Walters et al. [237]	Java	review, modification	eye-tracker, success	4 students, 4 practitioners
Busjahn et al. [28]	Java	reading, comprehension	eye-tracker	15 students
Busjahn et al. [29]	Java	comprehension	eye-tracker	14 students, 6 practitioners
Kevic et al. [120]	Java	review, modification	eye-tracker	10 students, 12 practitioners
Yusuf et al. [250]	UML	comprehension, modification	eye-tracker	12 students
Sharif and Maletic [198]	UML	comprehension	time, success	17 students
Sharif et al. [195, 197]	UML	comprehension	eye-tracker, time, success	15 students
Smet et al. [209]	UML	comprehension, modification	eye-tracker, time, success	23 students
Petrusel and Mendling [168]	BPMN	comprehension	eye-tracker	26 practitioners
Cagiltay et al. [31]	ER	comprehension, review	eye-tracker, time, success	9 practitioners
Sharafi et al. [192]	Tropos	comprehension	eye-tracker, time, success	28 students
Ikutani and Uwano [105]	source code	comprehension	NIRS	11 students
Siegmund et al. [204]	Java	comprehension	fMRI	17 students
Huang et al. [100]	data structures	comprehension	fNIRS, fMRI	76 students
Parnin [166]	source code	modification	EMG	2 participants
Fritz et al. [68]	C#	reading, comprehension	eye-tracker, EEG, EDA, time, success	15 practitioners
Störrle et al. [217]	UML	reading	eye-tracker	29 students
Müller and Fritz [147, 148]	Java	modification	eye-tracker, EEG, EDA	11 students, 6 practitioners
Çakir et al. [33]	Unity	creation, gender	success, time	21 children
Bruckman et al. [24]	MOOSE	creation, gender	success, time	475 children
Papavlasopoulou et al. [165]	Scratch	creation, gender	eye-tracker	149 children
Beckwith et al. [9, 10]	spreadsheets	review, gender	success	51 students
Torkzadeh and Koufteros [229]	source code	comprehension, gender	success	224 students

continue on next page...

Table 9.1: *...continued from previous page*

Study	Models/Code	Goal/Evaluation	Spec. language/ (Bio)metrics	Participants
Fisher et al. [64]	Java	comprehension, gender	success, time	30 students
Gramß et al. [85]	SysML, C	creation, gender	success, time	285 students
Sharafi et al. [191]	C	review, gender	eye-tracker, time, success	15 students
Hou et al. [99]	C	review, gender	eye-tracker success	25 students
Obaidellah and Haek [158]	pseudocode	comprehension	eye-tracker	51 students

9.5 Summary

In this Chapter, we started by presenting studies that evaluate the quality of requirements models through the analysis of differente properties of the models themselves. To do so, the majority of the analysed research work uses a set of metrics, some of them in combination with OCL.

Although it is important to have information about the model, we argue that information is not enough to have a complete perspective on its quality. As such, there is a need to investigate the impact of a given model on different activities performed by stakeholders, by exploring human factors and gender differences. When investigating these human factors, there are two main directions in the literature: the use of paper or web-based questionnaires, and the use of biometrics. In the latter, we noticed that most of the available research does not evaluate requirements models, but source code instead. In terms of requirements models, UML has been the main subject of research. Other models were fairly unexplored. Furthermore, to the best of our knowledge, the combination of all the (bio)metrics presented in this dissertation has not yet been explored, in the time of this writing. We concluded the Chapter by discussing the differences between the described research and this dissertation.

Chapter 10

Conclusions

In this closing Chapter, we revise the research work presented in this dissertation, and answer to the research questions presented Chapter 1. We reflect on the main contributions, as well as on the limitations. We further share lessons learnt from building and conducting several quasi-experiments involving human subjects and biometric devices. However, the journey through the quality evaluation of requirements models does not end here, thus we finish with directions for future work.

10.1 Answering the Research Questions and Contributions

RE approaches are used, among others, to facilitate the communication between requirements engineers and other stakeholder. However, communication flaws are among the most frequently reported RE problems that may led to software projects failures. One of the crucial elements of an effective communication is the quality of the requirements models used. However, RE approaches are still experiencing problems when it comes to managing the quality of the requirements models. As such, the general research question of this dissertation is:

How can we leverage a mixed-method process to characterise the quality of requirements models and the way stakeholders interact with them?

To answer these question, we proposed a mixed-method process named QualitEva. It is a step-by-step guide on how to perform (quasi-)experiments involving human subjects, and with the usage of (bio)metrics. It focuses on the particular case of using (quasi-) experiments to evaluate the quality of requirements models through both the analysis of the models themselves, and the exploitation of human factors on how different people interact with them.

Contribution 1: A generic mixed-method process for the quality evaluation of requirements models, which can be applied to various requirements models and quality characteristics (presented in Chapter 3).

We also propose a set of (bio)metrics, for performing the quality evaluation of requirements models based on the QualitEva process. The (bio)metrics are related with the evaluation of the *(i)* accuracy achieved by stakeholders when performing tasks on requirements models, as well as their *(ii)* speed *(iii)* visual ease; *(iv)* mental ease; *(v)* emotional ease; and *(vi)* perceived effort. We further propose metrics for the complexity and completeness evaluation of *(vii)* *i** models.

Contribution 2: A set of (bio)metrics for the evaluation of requirements models and the way stakeholders interact with them (presented in Chapter 4).

Contribution 3: Two online modelling and measurement tools, which automatically collect metrics about *i** 1.0 and iStar 2.0 models' complexity and completeness (presented in Chapter 4).

Our research on using a mixed-method process can be further divided into a more specific research question:

How can (bio)metric measurements be used to understand whether tasks such as creating, modifying, understanding and reviewing requirements models are difficult or easy to perform by a given stakeholder?

We also defined the null (H_0) and alternative (H_1) hypotheses, as follows:

$\boldsymbol{H_0}$ (Bio)metric measurements **do not** provide reliable quantitative information about the success and effort a stakeholder experiences while working on different requirements models' tasks.

$\boldsymbol{H_1}$ (Bio)metric measurements provide reliable quantitative information about the success and effort a stakeholder experiences while working on different requirements models' tasks.

To answer this questions and verify the hypothesis, we applied the QualitEva approach to the particular case of evaluating the appropriateness recognisability and learnability of *i** 1.0, iStar 2.0, ARNE use case template and ALCO use case template. We measured the accuracy, speed and ease of a total of 660 participants while performing creation, modification, understanding, or review tasks on these models. We used a combination of measurements, including metrics of task success, speed, and effort, collected with eye-tracking, EEG and EDA sensors, participants' perceived effort through a NASA-TLX questionnaire, and performed a characterisation of participants with GenderMag.

For *i**, our results indicate that participants were able to achieve a better performance and lower overall effort when using iStar 2.0 than the participants using *i** 1.0. Furthermore, when analysing the metrics about the *i** models created and modified by our participants, we were also able to conclude that using iStar 2.0 produced models with a lower accidental complexity than *i** 1.0, for the same problem description (hence, with the same essential complexity). For use cases, our results show that participants were able to achieve a better performance and lower overall effort when using ALCO than the participants using ARNE. In terms of participants characterisation, the results showed that participants with a comprehensive information processing style and a more conservative attitude towards risk (characteristics more frequently seen in females) took longer to start performing the tasks but had a higher accuracy. The visual effort, attention and mental workload was also higher for these participants.

Contribution 4: A quantitative evaluation providing empirical evidence on the usability of *i**, iStar 2.0, ARNE use case template, and ALCO use case template, in terms of appropriateness recognisability and learnability, by using a combination of (bio)metrics, in the tasks of creating, modifying, understanding and reviewing those models (presented in Chapters 5, 6, and 7).

When comparing textual and diagrammatic representation of requirements, our results indicate that participants using the latter have a higher precision and a lower overall effort. In particular, the results were better for ALCO use case template, followed by ARNE, iStar 2.0, and *i** 1.0. Although natural language can be potentially vague and prone to omissions and ambiguity, the participants are accustomed with it, being easier for them to understand it and perform the tasks. The difference in effort is particularly noticeable in participants with a comprehensive information processing style and a more conservative attitude towards risk (characteristics more frequently seen in females).

Contribution 5: A quantitative evaluation providing empirical evidence on the differences between *i** and use cases, in the tasks of creating, modifying, understanding and reviewing those models (presented in Chapter 8).

With all the collected data and achieved results, we are confident to reject the null hypothesis and state that (bio)metric measurements **provide** reliable quantitative information about the success and effort a stakeholder experiences while working on different requirements models' tasks.

To further validate our results, we encourage independent replications by international researchers. This is done by providing a a web-based replication package for the learnability and appropriateness recognisability of *i** 1.0, iStar 2.0, ARNE use cases, and ALCO use cases. It can easily be extended to facilitate the evaluation of other quality characteristics and different software artefacts, and it does not require any additional installation or computer software besides a web browser.

Contribution 6: An online replication package with all the materials used in the quasi-experiments reported in this dissertation, for facilitating independent replications. (presented in Chapter 5).

10.2 Limitations

In this dissertation, we are only focusing on usability of requirements models, one of the 8 quality characteristics according to the ISO/IEC FDSI 25023:2016 [107]. Furthermore, we are only addressing appropriateness recognisability and learnability, two of 6 sub-characteristics of usability. However, the QualitEva process proposed in this dissertation can be applied to evaluate different quality characteristics of requirements models, including the ones that are not part of this research work.

We are also aware that the size of both the *i** models and the use case descriptions used in quasi-experiments performed may not be representatives of the ones used in a real-work scenario. However, we were limited by the technical specifications of the eye-tracker device used, such as constraints in the external monitor dimensions and in the participant distance to the eye-tracker. As such, the materials should be readable by all participants, without the need for them to move closer to the screen, nor scroll down/up and zoom-in/out on the tasks.

Our quasi-experiments only cover one domain, a booking management system. The choice of a relatively known domain was well thought, as our goal was to reduce the effect of the results being related with difficulties in understanding the domain itself, and not due to the requirements languages that were being studied. However, this choice has the consequence of tacit knowledge influencing the results. The domain selection is, in fact, a *double-edged sword*. Since we intended to evaluate the requirements languages, reducing confounding effects was considered a priority.

10.3 Lessons Learnt

There are some lessons we learnt from building and conducting several quasi-experiments involving human subjects and biometric devices. We describe them as a way to help other researchers and in the spirit of open science and knowledge sharing. They are not tied to the evaluation of requirements models in particular, rather to the general process of conducting (quasi-)experiments in Software Engineering. As such, we decided to place them in the Conclusions of this dissertation, and not as part of a specific Chapter.

The instrumentation of the experiments takes time and it is error prone.
Planning a (quasi-)experiment is a laborious task, since several aspects need to be taken into consideration, as detailed in Chapter 3. However, even with a well-defined process and a carefully planned experiment, the instrumentation itself can take longer than anticipated. Be prepared to create the experimental materials and change them several times before the experiments can be conducted. These changes can be caused by simple typos,

or even by a major restructuring of the materials to accommodate a specific formalism of the requirements model that was not considered before. Furthermore, when using an eye-tracker with screen size limitations, the size of the fonts or model elements may also need to be changed. When creating the video tutorials, we were using presentation slides and a text to speech tool. The text had some typos, that were only noticed when the recording was taking place, meaning that we needed to correct them and start all over again with the recording. This can happen even when carefully reading all the materials. Therefore, we recommend using an external person to proofread everything, since (s)he will read (or listen) the experimental materials from an objective perspective and can easily catch small errors that we missed. Additionally, conducting a pilot study, as described in Subsection 3.3.9, is also paramount to understand if all the experimental materials, including chairs, and computer peripherals, are adequate. We recommend to consider at least 2-3 months to prepare all the materials, conduct the pilot study, and perform the corresponding changes.

The setup and disassemble of an experiment takes time.
For conducting our experiments, we were using a computer, an external monitor, and external mouse and keyboard, and the 3 biometric devices (eye-tracker, EEG and EDA scanners). To connect all these equipments, we also needed a hub USB and an extension cord. Furthermore, in the quasi-experiments conducted at software companies, we had to take all the equipment with us, assemble them before the studies, and disassemble them in the end. Sometimes, we also needed to change the positions of the chairs, and the light in the room. Furthermore, we had to test the eye-tracker with the light and change the window blinds accordingly. This initial setup took 20-25 minutes, on average. Furthermore, in the day before of the experiments, we needed to make sure that the EEG scanner batteries had power and that the EDA was charged.

Biometric sensors can and will fail during an experimental session.
The devices we used in the quasi-experiments are low-cost and non-invasive biosensors. To gain this non-intrusive property, the EEG and EDA devices have to use a wireless connection to communicate with the computer, namely Bluetooth. Although not very common, interferences with the signal do happen, and may cause the devices to lose connection with the computer. This results in the exclusion of the corresponding EEG or EDA data for that participant. We can minimise possible interferences by turning off Wi-Fi and other devices working on the same 2.4 GHz frequency, if not needed to perform the experiment.

Collaboration with industry is easier when you have an undercover agent inside.
From our experience contacting companies in Portugal, Canada, and USA, leveraging personal contacts is the best and easiest way to be able to work with software companies and enter their offices. Personal contacts can indicate the person with whom we need to talk about performing experiments with the employees, or can even talk directly with that person. In both cases, it is important to have a summary of experiment, and guarantee that the normal work day will not be greatly affected by our presence.

Some people will be impressed by the biosensors. Others, will need reassurance. Having people skills and knowing how to interpret a reaction can be fundamental when running (quasi)-experiments with human subjects. We noticed that several participants were interested in the biosensors and wanted to know more about them, and even if they could buy one to make some experiments at home. Others, however, were afraid of the devices and the type of information they could collect. This apprehension could jeopardise the results, and cause the participant to be stressed about the biodata collected and not the task (s)he had to perform. As such, explaining how the devices work (but not explicitly informing what was being tested, as it could introduce another threat), and communicating how the anonymity would be guaranteed, relaxed the participants.

People will ask questions, even when they are instructed not to do so. In the beginning of the quasi-experiments, we informed the participants that they should control the entire session, no question would be answered, no feedback was going to be provided, and they should behave as if no one else was in the room. In an ideal situation, we would have left the participant alone to perform the tasks. However, we needed to guarantee that everything was working during the course of the experiment. Nonetheless, we were as far away of the participant as possible, and tried to minimise the effects of our presence. Even so, several participants asked questions like "Am I replying this correctly?". This is part of the human behaviour, and it is hard to control.

People will lie and omit important information, if the opportunity is given. When asking to evaluate a given artefact, it is common for people to try to please and give an answer they think the experimenter wants. Although we have not observed this behaviour with NASA-TLX, it was evident in the video recording of the answers to the demographic questionnaire that people do lie and omit information. When they have to access memory and remember how long ago they used a specific requirements model, several participants tried to remember and selected a few options, but then decided to say it was their first time with using it. We had to use the answers saved in the questionnaire, and not the other options the participants selected before. In previous quasi-experiments, when we have not given examples of the answers to the participants, they would simply ignore the question or answer "I don't known". As such, we tried to minimise this by always giving examples of what we expected the answer to be, and giving specific time intervals for questions involving temporal durations. Even for nationality, we had "e.g., Portuguese". This way, people will see a possible answer and better understand what is asked, reducing the chances of not replying to the question.

10.4 Future Work

We plan to continue working on the topics covered in this dissertation for the years to come. The discussion on future work can be organised into two main components: *(i)* different types of analyses for the data already collected in the performed quasi-experiments, and *(ii)* further evaluations and lines of research.

10.4.1 Further Analysis on the Collected Data

In the quasi-experiments presented in this dissertation, we collected data from 660 participants with different personal and professional characteristics. Although we have performed a comprehensive analysis of the data, there are still several types of studies that can be performed with the data already collected, without conducting further quasi-experiments. We explore a few here.

We are interested in studying the differences in the results achieved based on the experience of the participants. The objective is to compare the differences between students, working students, practitioners and researchers. Students can be further refined into BSc, MSc or PhD students, while practitioners and researchers can be classified as junior or senior. Furthermore, we also interested in analysing the differences in the results achieved based on the field of studies and work of the participants. These analyses can give us insights on the impact of the professional experience and background when performing tasks on requirements models.

The information collected from the biometric sensors used in this dissertation is very rich. With the data we already have, there are several types of analysis that can be performed and that were not explored. Specifically in terms of eye-data from the eye-tracker, we intend to analyse gaze-plots and scan paths, in order to identify navigation patterns. The goals are, among others, to explore if the way a participant navigates through a model depends on the question, and if the success achieved in the task can be related with different navigation strategies. With the brain-data from the EEG, we aim to explore the level of innovative and creative thinking a participant experiences while performing the tasks. Finally, with the skin- and heart-data from the EDA, the objective is to perform a detailed analysis on the emotional spectrum. All these investigations should take into consideration the individual characteristics of the participants, obtained from the GenderMag characterisation.

10.4.2 Further Evaluations and Lines of Research

There are some other directions related with the quality of requirements models that we would like to explore in the future. Firstly, it is necessary to assess how consistently the results achieved in this dissertation occur with other users, problem descriptions, and models. We plan to replicate the experiment in other contexts, and apply it to bigger and more complex descriptions. However, we also encourage independent replications. We are also interested in applying the same techniques used in this dissertation to other sub-characteristics of usability, namely accessibility from an inclusiveness perspective.

For the case of iStar 2.0, we plan to perform a study on the evolution and ease of learning of the language, by analysing students' projects. Ideally, the works would be divided into phases, and the models created in the different phases would be analysed, with an evaluation of their evolution. This would give us important information regarding which are the most common mistakes and difficulties when learning iStar 2.0. Furthermore,

we are also interested in creating and evaluating a timely feedback mechanism for requirements engineering, concerning the quality of the requirements models they create, while they are being created. That is, a presentation of metrics and suggestions in the requirements modelling tool, and the corresponding analysis of benefits and disadvantages of such feedback. Furthermore, a post-mortem analysis can also be performed, by having a log of the model creation and modification process. As such, we envision an experiment where one group would have the access to the metrics and corresponding feedback available, and another would only have the tool.

For bigger and more complex requirements descriptions, we are interested in evaluating use cases with alternative scenarios and includes/extends relationships. Backward and forward traceability of requirements is an important issue, which we would like to further explore. We are also interested in analysing completeness and soundness of requirements. For *i**, we are interested in studying the usage of both SD and SR models in the ability to understand the models.

Other lines of research that we are interested in, include the usage of machine learning techniques to predict navigation patterns while a stakeholder is performing different types of tasks on requirements models. Machine learning has been shown to be a good approach for finding links between low-level biometric data and high-level phenomena, such as perceived difficulty and quality concerns [13]. Furthermore, we plan to apply the same techniques used in this dissertation to other domains, namely Computer Science education and training. We argue that, by improving the ways we teach, we will be able to have better professionals and, ultimately, better software.

Several other evaluations and studies can be pursued in the future. New ideas will appear from the ones presented here, and each experiment will bring us closer to our ultimate goal: improve the process of building better software, with higher quality standards and that addresses real needs.

Bibliography

[1] T. Adlin and J. Pruitt. *The Essential Persona Lifecycle: Your Guide to Building and Using Personas*. 1st ed. Morgan Kaufmann, 2010. ISBN: 9780123814180.

[2] B. Anda, D. Sjøberg, and M. Jørgensen. "Quality and Understandability of Use Case Models." In: *Proceedings of the 16th European Conference on Object-Oriented Programming (ECOOP 2001)*. Springer, 2001, pp. 402–428. DOI: 10.1007/3-540-45337-7_21.

[3] J. L. Andreassi. *Psychophysiology: Human Behavior & Physiological Response*. Psychology Press, 2013. ISBN: 978-1-135-68119-7.

[4] L. Anthony, P. Carrington, P. Chu, C. Kidd, J. Lai, and A. Sears. "Gesture Dynamics: Features Densitive to Task Difficulty and Correlated with Physiological Sensors." In: *Stress* 1418.360 (2011), pp. 312–316.

[5] J. Arlow and I. Neustadt. *UML 2 and the Unified Process: Practical Object-oriented Analysis and Design*. 2nd ed. Pearson Education, 2005. ISBN: 978-0321321275.

[6] A. L. Baroni, S. Braz, and F. B. e Abreu. "Using OCL to Formalize Object-Oriented Design Metrics Definitions." In: *Proceedings of the 6th European Conference on Object-Oriented Programming Workshop on Quantitative Aspects of Object-Oriented Software Engineering (QAOOSE 2002)*. Springer-Verlag, 2002.

[7] V. R. Basili and H. D. Rombach. "The TAME Project: Towards Improvement-Oriented Software Environments." In: *IEEE Transactions on Software Engineering* 14.6 (1988), pp. 758–773. DOI: 10.1007/3-540-27662-9_8.

[8] V. R. Basili, G. Caldiera, and H. D. Rombach. "Goal Question Metric Paradigm." In: *Encyclopedia of Software Engineering*. Ed. by J. J. Marciniak. 1st ed. New York, NY, USA: Wiley, 2001. ISBN: 1-54004-8.

[9] L. Beckwith, M. Burnett, S. Wiedenbeck, C. Cook, S. Sorte, and M. Hastings. "Effectiveness of End-User Debugging Software Features: Are There Gender Issues?" In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2005)*. ACM. 2005, pp. 869–878. DOI: 10.1145/1054972.1055094.

[10] L. Beckwith, C. Kissinger, M. Burnett, S. Wiedenbeck, J. Lawrance, A. Blackwell, and C. Cook. "Tinkering and Gender in End-User Programmers' Debugging." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2006)*. ACM. 2006, pp. 231–240. DOI: 10.1145/1124772.1124808.

[11] R. Bednarik and M. Tukiainen. "An Eye-tracking Methodology for Characterizing Program Comprehension Processes." In: *Proceedings of the ACM Symposium on Eye Tracking Research & Applications (ETRA 2006)*. ACM. 2006, pp. 125–132. DOI: 10.1145/1117309.1117356.

[12] R. Bednarik and M. Tukiainen. "Temporal Eye-Tracking Data: Evolution of Debugging Strategies with Multiple Representations." In: *Proceedings of the ACM Symposium on Eye Tracking Research & Applications (ETRA 2008)*. ACM. 2008, pp. 99–102. DOI: 10.1145/1344471.1344497.

[13] R. Bednarik, H. Vrzakova, and M. Hradis. "What Do You Want to Do Next: A Novel Approach for Intent Prediction in Gaze-based Interaction." In: *Proceedings of the Symposium on Eye Tracking Research and Applications*. ACM. 2012, pp. 83–90. DOI: 10.1145/2168556.2168569.

[14] B. Berenbach and G. Borotto. "Metrics for Model Driven Requirements Development." In: *Proceedings of the 28th International Conference on Software Engineering (ICSE 2006)*. ACM. 2006, pp. 445–451. DOI: 10.1145/1134285.1134348.

[15] C. Berka, D. J. Levendowski, M. N. Lumicao, A. Yau, G. Davis, V. T. Zivkovic, R. E. Olmstead, P. D. Tremoulet, and P. L. Craven. "EEG Correlates of Task Engagement and Mental Workload in Vigilance, Learning, and Memory Tasks." In: *Aviation, Space, and Environmental Medicine* 78.5 (2007), B231–B244.

[16] T. Besker, A. Martini, R. E. Lokuge, K. Blincoe, and J. Bosch. "Embracing Technical Debt, from a Startup Company Perspective." In: *Proceedings of the IEEE International Conference on Software Maintenance and Evolution (ICSME 2018)*. IEEE, 2018, pp. 415–425. DOI: 10.1109/ICSME.2018.00051.

[17] BioSignalsPlux Wristband. *Wearable Body Sensing Platform*. (Last access: September 2019). URL: https://biosignalsplux.com/.

[18] BITalino. (Last access: September 2019). URL: http://bitalino.com/.

[19] D. Bombonatti, C. Gralha, A. Moreira, J. Araújo, and M. Goulão. "Usability of Requirements Rechniques: A Systematic Literature Review." In: *Proceedings of the 31st Annual ACM Symposium on Applied Computing (SAC 2016)*. ACM. 2016, pp. 1270–1275. DOI: 10.1145/2851613.2851758.

[20] G. Booch, R. A. Maksimchuk, M. W. Engle, B. J. Young, J. Conallen, and K. A. Houston. *Object-Oriented Analysis and Design with Applications*. 3rd ed. USA: Addison-Wesley Professional, 2007. ISBN: 978-0201895513.

[21] W. Boucsein. *Electrodermal Activity*. 2nd ed. Springer Science & Business Media, 2012. ISBN: 978-1-4614-1126-0. DOI: 10.1007/978-1-4614-1126-0.

[22] J. B. Brookings, G. F. Wilson, and C. R. Swain. "Psychophysiological Responses to Changes in Workload During Simulated Air Traffic Control." In: *Biological Psychology* 42.3 (1996), pp. 361–377. DOI: 10.1016/0301-0511(95)05167-8.

[23] F. P. Brooks Jr. *The Mythical Man-Month: Essays on Software Engineering (Anniversary Edition)*. 2nd ed. Addison-Wesley, 2015. ISBN: 978-0201835953.

[24] A. Bruckman, C. Jensen, and A. de Bonte. "Gender and Programming Achievement in a CSCL Environment." In: *Proceedings of the Conference on Computer Support for Collaborative Learning: Foundations for a CSCL Community (CSCL 2002)*. International Society of the Learning Sciences, 2002, pp. 119–127. URL: http://dl.acm.org/citation.cfm?id=1658634.

[25] A. S. Bruckman. "MOOSE Crossing: Construction, Community and Learning in a Networked Virtual World for Kids." Doctoral dissertation. Massachusetts Institute of Technology, 1997.

[26] M. Burnett, A. Horvath, and A. Oleson. *GenderMag Personas Foundations Document*. (Last access: September 2019). URL: http://eusesconsortium.org/gender/GenderMagPersona-FoundationDocuments/Foundations.html.

[27] M. Burnett, S. Stumpf, J. Macbeth, S. Makri, L. Beckwith, I. Kwan, A. Peters, and W. Jernigan. "GenderMag: A Method for Evaluating Software's Gender Inclusiveness." In: *Interacting with Computers* 28.6 (2016), pp. 760–787. DOI: 10.1093/iwc/iwv046.

[28] T. Busjahn, R. Bednarik, and C. Schulte. "What Influences Dwell Time During Source Code Reading? Analysis of Element Type and Frequency as Factors." In: *Proceedings of the ACM Symposium on Eye Tracking Research & Applications (ETRA 2014)*. ACM. 2014, pp. 335–338. DOI: 10.1145/2578153.2578211.

[29] T. Busjahn, R. Bednarik, A. Begel, M. Crosby, J. H. Paterson, C. Schulte, B. Sharif, and S. Tamm. "Eye Movements in Code Reading: Relaxing the Linear Order." In: *Proceedings of the IEEE 23rd International Conference on Program Comprehension (ICPC 2015)*. IEEE. 2015, pp. 255–265. DOI: 10.1109/ICPC.2015.36.

[30] J. P. Byrnes, D. C. Miller, and W. D. Schafer. "Gender Differences in Risk Taking: A Meta-analysis." In: *Psychological Bulletin* 125.3 (1999), pp. 367–383. DOI: 10.1037/0033-2909.125.3.367.

[31] N. E. Cagiltay, G. Tokdemir, O. Kilic, and D. Topalli. "Performing and Analyzing Non-formal Inspections of Entity Relationship Diagram (ERD)." In: *Journal of Systems and Software* 86.8 (2013), pp. 2184–2195. DOI: 10.1016/j.jss.2013.03.106.

[32] P. Caire, N. Genon, P. Heymans, and D. L. Moody. "Visual Notation Design 2.0: Towards User Comprehensible Requirements Engineering Notations." In: *Proceedings of the 21st IEEE International Requirements Engineering Conference (RE 2013)*. 2013, pp. 115–124. DOI: 10.1109/RE.2013.6636711.

[33] N. A. Çakir, A. Gass, A. Foster, and F. J. Lee. "Development of a Game-design Workshop to Promote Young Girls' Interest Towards Computing Through Identity Exploration." In: *Computers & Education* 108 (2017), pp. 115–130. DOI: 10.1016/j.compedu.2017.02.002.

[34] N. R. Carlson. *Physiology of Behavior*. 12th ed. Pearson, 2019. ISBN: 9780205239399.

[35] E. Carniglia, M. Caputi, V. Manfredi, D. Zambarbieri, and E. Pessa. "The Influence of Emotional Picture Thematic Content on Exploratory Eye Movements." In: *Journal of Eye Movement Research* 5.4 (2012). DOI: 10.16910/jemr.5.4.4.

[36] A. J. Casson, D. C. Yates, S. J. M. Smith, J. S. Duncan, and E. Rodriguez-Villegas. "Wearable EEG: What is it, Why is it Needed, and What Does it Entail?" In: *IEEE Engineering in Medicine and Biology Magazine* 29.3 (2010), pp. 44–56. DOI: 10.1109/IEMBS.2008.4650549.

[37] J. Castro, M. Kolp, and J. Mylopoulos. "A Requirements-Driven Development Methodology." In: *Proceedings of the 13th International Conference on Advanced Information Systems Engineering (CAiSE 2001)*. Springer, 2001, pp. 108–123. DOI: 10.1007/3-540-45341-5_8.

[38] K. Chari and M. Agrawal. "Impact of Incorrect and New Requirements on Waterfall Software Project Outcomes." In: *Empirical Software Engineering* 23.1 (2018), pp. 165–185. DOI: 10.1007/s10664-017-9506-4.

[39] M. Chemuturi. *Requirements Engineering and Management for Software Development Projects*. 1st ed. Springer Science & Business Media, 2013. ISBN: 978-1-4614-5377-2. DOI: 10.1007/978-1-4614-5377-2.

[40] S. Cheryan, S. A. Ziegler, A. K. Montoya, and L. Jiang. "Why Are Some STEM Fields More Gender Balanced than Others?" In: *Psychological Bulletin* 143.1 (2017), p. 1. DOI: 10.1037/bul0000052.

[41] M. B. Chrissis, M. Konrad, and S. Shrum. *CMMI Guidlines for Process Integration and Product Improvement*. 3rd ed. Addison-Wesley Professional, 2011. ISBN: 978-0321711502.

[42] A. Cockburn. *Writing Effective Use Cases: The Crystal Collection for Software Professionals*. 1st ed. Addison-Wesley Professional Reading, 2000. ISBN: 978-0201702255.

[43] J. Cohen. "A Power Primer." In: *Psychological Bulletin* 112.1 (1992), pp. 155–159. DOI: 10.1037/0033-2909.112.1.155.

[44] N. Condori-Fernandez, M. Daneva, K. Sikkel, R. Wieringa, O. Dieste, and O. Pastor. "A Systematic Mapping Study on Empirical Evaluation of Software Requirements Specifications Techniques." In: *Proceedings of the 3rd International Symposium on Empirical Software Engineering and Measurement (ESEM 2009)*. IEEE Computer Society. 2009, pp. 502–505. DOI: 10.1109/ESEM.2009.5314232.

[45] T. D. Cook and D. T. Campbell. *Quasi-Experimentation: Design and Analysis Issues for Field Settings*. 1st ed. Houghton Mifflin, 1979. ISBN: 978-0395307908.

[46] J. Corbin and A. Strauss. *Basic of Qualitative Research: Techniques and Procedures for Developing Grounded Theory*. 4th ed. SAGE Publications, 2014. ISBN: 978-1412997461.

[47] M. E. Crosby and J. Stelovsky. "How Do We Read Algorithms? A Case Study." In: *IEEE Computer* 23.1 (1990), pp. 25–35. DOI: 10.1109/2.48797.

[48] M. Crowne. "Why Software Product Startups Fail and What to do About It. Evolution of Software Product Development in Startup Companies." In: *Proceedings of the IEEE International Engineering Management Conference*. Vol. 1. IEEE. 2002, pp. 338–343. DOI: 10.1109/IEMC.2002.1038454.

[49] F. Dalpiaz, X. Franch, and J. Horkoff. "iStar 2.0 Language Guide." In: *CoRR* abs / 1605.07767 (2016). (Last access: September 2019). URL: https://arxiv.org/abs/1605.07767v3.

[50] J. Daly, A. Brooks, J. Miller, M. Roper, and M. Wood. "Evaluating Inheritance Depth on the Maintainability of Object-Oriented Software." In: *Empirical Software Engineering* 1.2 (1996), pp. 109–132. DOI: 10.1007/BF00368701.

[51] J. Daniel. *Sampling Essentials: Practical Guidelines for Making Sampling Choices*. 1st ed. SAGE Publications, 2011. ISBN: 978-1412952217.

[52] A. M. Davis. *Software Requirements: Objects, Functions, and States*. Prentice-Hall, Inc., 1993. ISBN: 0-13-805763-X.

[53] J. M. C. De Gea, J. Nicolás, J. L. F. Alemán, A. Toval, C. Ebert, and A. Vizcaíno. "Requirements Engineering Tools: Capabilities, Survey and Assessment." In: *Information and Software Technology* 54.10 (2012), pp. 1142–1157. DOI: 10.1016/j.infsof.2012.04.005.

[54] R. K. Dishman, Y. Nakamura, M. E. Garcia, R. W. Thompson, A. L. Dunn, and S. N. Blair. "Heart rate variability, trait anxiety, and perceived stress among physically fit men and women." In: *International Journal of Psychophysiology* 37.2 (2000), pp. 121–133. DOI: 10.1016/S0167-8760(00)00085-4.

[55] A. Duchowski. *Eye Tracking Methodology: Theory and Practice*. 2nd ed. Vol. 373. Springer Science & Business Media, 2007. ISBN: 978-1-84628-609-4.

[56] H. Eichelberger. "Aesthetics of Class Diagrams." In: *Proceedings of the 1st International Workshop on Visualizing Software for Understanding and Analysis (VISSOFT 2002)*. IEEE. 2002, pp. 23–31. DOI: 10.1109/VISSOF.2002.1019791.

[57] H. Eichelberger. "Nice class diagrams admit good design?" In: *Proceedings of the ACM symposium on Software visualization (SOFTVIS 2013)*. ACM. 2003, pp. 159–167. DOI: 10.1145/774833.774857.

[58] P. Ekman, R. W. Levenson, and W. V. Friesen. "Autonomic Nervous System Activity Distinguishes Among Emotions." In: *Science* 221.4616 (1983), pp. 1208–1210. DOI: 10.1126/science.6612338.

[59] K. El Emam, S. Benlarbi, N. Goel, and S. N. Rai. "The Confounding Effect of Class Size on the Validity of Object-Oriented Metrics." In: *IEEE Transactions on Software Engineering* 27.7 (2001), pp. 630–650. DOI: 10.1109/32.935855.

[60] A. Endres and H. D. Rombach. *A Handbook of Software and Systems Engineering: Empirical Observations, Laws, and Theories*. 1st ed. Pearson Education, 2003. ISBN: 978-0321154200.

[61] J. Engmann. "Evaluating the Impact of Evolving Requirements on Wider System Goals: Using i* Methodology Integrated with Satisfaction Arguments to Evaluate the Impact of Changing Requirements in HIV/AIDS Monitoring Systems in the UK." Master's thesis. England: School of Informatics, City University of London, 2009.

[62] P. Espada, M. Goulão, and J. Araújo. "A Framework to Evaluate Complexity and Completeness of KAOS Goal Models." In: *Proceeding of the 25th International Conference on Advanced Information Systems Engineering (CAiSE 2013)*. Springer. 2013, pp. 562–577. DOI: 10.1007/978-3-642-38709-8_36.

[63] D. Falessi, N. Juristo, C. Wohlin, B. Turhan, J. Münch, A. Jedlitschka, and M. Oivo. "Empirical Software Engineering Experts on the Use of Students and Professionals in Experiments." In: *Empirical Software Engineering* 23.1 (2018), pp. 452–489. DOI: 10.1007/s10664-017-9523-3.

[64] M. Fisher, A. Cox, and L. Zhao. "Using sex differences to link spatial cognition and program comprehension." In: *Proceedings of the 22nd IEEE International Conference on Software Maintenance (ICSM 2006)*. IEEE. 2006, pp. 289–298. DOI: 10.1109/ICSM.2006.72.

[65] M. Fowler and J. Highsmith. "The Agile Manifesto." In: *Software Development* 9.8 (2001), pp. 28–35. DOI: 10.1007/978-3-319-10157-6_3.

[66] X. Franch. "A Method for the Definition of Metrics over i* Models." In: *Proceeding of the 21st International Conference on Advanced Information Systems Engineering (CAiSE 2009)*. Springer. 2009, pp. 201–215. DOI: 10.1007/978-3-642-02144-2_19.

[67] X. Franch and G. Grau. "Towards a Catalogue of Patterns for Defining Metrics Over i* Models." In: *Proceeding of the 20th International Conference on Advanced Information Systems Engineering (CAiSE 2008)*. Springer. 2008, pp. 197–212. DOI: 10.1007/978-3-540-69534-9_16.

[68] T. Fritz, A. Begel, S. C. Müller, S. Yigit-Elliott, and M. Züger. "Using Psycho-physiological Measures to Assess Task Difficulty in Software Development." In: *Proceedings of the 36th International Conference on Software Engineering (ICSE 2014)*. ACM, 2014, pp. 402–413. DOI: 10.1145/2568225.2568266.

[69] S. Galhotra, Y. Brun, and A. Meliou. "Fairness Testing: Testing Software for Discrimination." In: *Proceedings of the 11th Joint Meeting on Foundations of Software Engineering*. ACM. 2017, pp. 498–510. DOI: 10.1145/3106237.3106277.

[70] D. Galin. *Software Quality Assurance: From Theory to Implementation*. 1st ed. Pearson Education, 2004. ISBN: 978-0201709452.

[71] A. Gevins, M. E. Smith, H. Leong, L. McEvoy, S. Whitfield, R. Du, and G. Rush. "Monitoring Working Memory Load During Computer-based Tasks With EEG Pattern Recognition Methods." In: *Human Factors: The Journal of the Human Factors and Ergonomics Society* 40.1 (1998), pp. 79–91. DOI: 10.1518/001872098779480578.

[72] C. Giardino, N. Paternoster, M. Unterkalmsteiner, T. Gorschek, and P. Abrahamsson. "Software Development in Startup Companies: The Greenfield Startup Model." In: *IEEE Transactions on Software Engineering* 42.6 (2016), pp. 585–604. DOI: 10.1109/TSE.2015.2509970.

[73] P. Giorgini, J. Mylopoulos, and R. Sebastiani. "Goal-oriented Requirements Analysis and Reasoning in the TROPOS Methodology." In: *Engineering Applications of Artificial Intelligence* 18.2 (2005), pp. 159–171. DOI: 10.1016/j.engappai.2004.11.017.

[74] GNU PSPP – Free Software Foundation. (Last access: September 2019). URL: https://www.gnu.org/software/pspp/.

[75] J. H. Goldberg and X. P. Kotval. "Computer Interface Evaluation Using Eye Movements: Methods and Constructs." In: *International Journal of Industrial Ergonomics* 24.6 (1999), pp. 631–645. DOI: 10.1016/S0169-8141(98)00068-7.

[76] H. Gomaa. *Software Modeling and Design: UML, Use Cases, Patterns, and Software Architectures*. 1st ed. Cambridge University Press, 2011. ISBN: 978-0521764148.

[77] O. S. Gómez, N. Juristo, and S. Vegas. "Understanding replication of experiments in software engineering: A classification." In: *Information & Software Technology* 56.8 (2014), pp. 1033–1048. DOI: 10.1016/j.infsof.2014.04.004.

[78] E. Goncalves, J. Castro, J. Araújo, and T. Heineck. "A Systematic Literature Review of iStar Extensions." In: *Journal of Systems and Software* 137 (2018), pp. 1–33. DOI: 10.1016/j.jss.2017.11.023.

[79] T. Gorschek, P. Garre, S. Larsson, and C. Wohlin. "A model for technology transfer in practice." In: *IEEE Software* 23.6 (2006), pp. 88–95. DOI: 10.1109/MS.2006.147.

[80] M. Goulão. "Component-Based Software Engineering: A Quantitative Approach." Doctoral dissertation. Portugal: Departamento de Informática, Universidade Nova de Lisboa, 2008.

[81] C. Gralha, M. Goulão, and J. Araújo. "Identifying Modularity Improvement Opportunities in Goal-oriented Requirements Models." In: *Proceeding of the 26th International Conference on Advanced Information Systems Engineering (CAiSE 2014)*. Springer. 2014, pp. 91–104. DOI: 10.1007/978-3-319-07881-6_7.

[82] C. Gralha, J. Araújo, and M. Goulão. "Metrics for Measuring Complexity and Completeness for Social Goal Models." In: *Information Systems* 53 (2015), pp. 346–362. DOI: 10.1016/j.is.2015.03.006.

[83] C. Gralha, D. Damian, A. Wasserman, M. Goulão, and J. Araújo. "The Evolution of Requirements Practices in Software Startups." In: *Proceedings of the 40th International Conference on Software Engineering (ICSE 2018)*. ACM. 2018, pp. 823–833. DOI: 10.1145/3180155.3180158.

[84] C. Gralha, M. Goulão, and J. Araújo. "Analysing Gender Differences in Building Social Goal Models: A Quasi-Experiment." In: *Proceedings of the IEEE 27th International Requirements Engineering Conference (RE 2019)*. IEEE. 2019, to appear.

[85] D. Gramß, T. Frank, S. Rehberger, and B. Vogel-Heuser. "Female Characteristics and Requirements in Software Engineering in Mechanical Engineering." In: *Proceedings of the International Conference on Interactive Collaborative Learning (ICL 2014)*. IEEE. 2014, pp. 272–279. DOI: 10.1109/ICL.2014.7017783.

[86] V. Grigoreanu, M. Burnett, S. Wiedenbeck, J. Cao, K. Rector, and I. Kwan. "End-user Debugging Strategies: A Sensemaking Perspective." In: *ACM Transactions on Computer-Human Interaction* 19.1 (2012), p. 5. DOI: 10.1145/2147783.2147788.

[87] D. Grimes, D. S. Tan, S. E. Hudson, P. Shenoy, and R. P. N. Rao. "Feasibility and pragmatics of classifying working memory load with an electroencephalograph." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2008)*. ACM. 2008, pp. 835–844. DOI: 10.1145/1357054.1357187.

[88] G. Guizzardi and T. P. Sales. "Detection, Simulation and Elimination of Semantic Anti-patterns in Ontology-driven Conceptual Models." In: *Proceedings of the 33rd International Conference on Conceptual Modeling (ER 2014)*. Springer, 2014, pp. 363–376. DOI: 10.1007/978-3-319-12206-9_30.

[89] A. Haag, S. Goronzy, P. Schaich, and J. Williams. "Emotion Recognition Using Bio-Sensors: First Steps Towards an Automatic System." In: *Proceedings of the Tutorial and Research Workshop on Affective Dialogue System (ASD 2004)*. Springer. 2004, pp. 36–48. DOI: 10.1007/978-3-540-24842-2_4.

[90] E. Haapalainen, S. Kim, J. F. Forlizzi, and A. K. Dey. "Psycho-physiological Measures for Assessing Cognitive Load." In: *Proceedins of the 12th ACM International Conference on Ubiquitous Computing (UbiComp 2010)*. ACM. 2010, pp. 301–310. DOI: 10.1145/1864349.1864395.

[91] P. A. Hancock and M. H. Chignell. "Toward a Theory of Mental Workload: Stress and Adaptability in Human-Machine Systems." In: *IEEE Transactions on Systems, Man and Cybernetics* (1986), pp. 378–383.

[92] T. C. Handy. *Event-related Potentials: A Methods Handbook*. The MIT press, 2005. ISBN: 978-0-26208-333-1.

[93] S. G. Hart. "NASA-Task Load Index (NASA-TLX); 20 Years Later." In: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*. Vol. 50. 9. SAGE Publications. 2006, pp. 904–908. DOI: 10.1177/154193120605000909.

[94] S. G. Hart and L. E. Staveland. "Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research." In: *Advances in Psychology* 52 (1988), pp. 139–183. DOI: 10.1016/S0166-4115(08)62386-9.

[95] S. G. Hill, H. P. Iavecchia, J. C. Byers, A. C. Bittner, A. L. Zaklade, and R. E. Christ. "Comparison of Four Subjective Workload Rating sScales." In: *Human Factors: The Journal of the Human Factors and Ergonomics Society* 34.4 (1992), pp. 429–439. DOI: 10.1177/001872089203400405.

[96] J. Horkoff and E. Yu. "Comparison and Evaluation of Goal-oriented Satisfaction Analysis Techniques." In: *Requirements Engineering* 18.3 (2013), pp. 199–222. DOI: 10.1007/s00766-011-0143-y.

[97] J. Horkoff, F. B. Aydemir, E. Cardoso, T. Li, A. Maté, E. Paja, M. Salnitri, J. Mylopoulos, and P. Giorgini. "Goal-Oriented Requirements Engineering: A Systematic Literature Map." In: *Proceedings of the IEEE 24th International Requirements Engineering Conference (RE 2016)*. IEEE. 2016, pp. 106–115. DOI: 10.1109/RE.2016.41.

[98] M. Höst, B. Regnell, and C. Wohlin. "Using Students as Subjects – A Comparative Study of Students and Professionals in Lead-time Impact Assessment." In: *Empirical Software Engineering* 5.3 (2000), pp. 201–214. DOI: 10.1023/A:1026586415054.

[99] T. Hou, Y. Lin, Y. Lin, C. Chang, and M. Yen. "Exploring the Gender Effect on Cognitive Processes in Program Debugging based on Eye-movement Analysis." In: *Proceedings of the 5th International Conference on Computer Supported Education (CSEDU 2013)*. 2013, pp. 469–473. DOI: 10.5220/0004415104690473.

[100] Y. Huang, X. Liu, R. Krueger, T. Santander, X. Hu, K. Leach, and W. Weimer. "Distilling Neural Representations of Data Structure Manipulation using fMRI and fNIRS." In: *Proceedings of the 41st International Conference on Software Engineering (ICSE 2019)*. IEEE. 2019, pp. 396–407. DOI: 10.1109/ICSE.2019.00053.

[101] E. Hull, K. Jackson, and J. Dick. *Requirements Engineering*. 3rd ed. Springer Science & Business Media, 2011. ISBN: 978-1849964043.

[102] i* 1.0 and iStar 2.0 modelling tools. (Last access: September 2019). URL: http://microlina.github.io/Framework/tools/tools.html.

[103] IBM SPSS Statistics. (Last access: September 2019). URL: https://www.ibm.com/products/spss-statistics.

[104] IEEE Computer Society. *1061-1998 – IEEE Standard for a Software Quality Metrics Methodology*. (Last access: September 2019). URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=749159.

[105] Y. Ikutani and H. Uwano. "Brain Activity Measurement During Program Comprehension with NIRS." In: *Proceedings of the 15th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD 2014)*. IEEE. 2014, pp. 1–6. DOI: 10.1109/SNPD.2014.6888727.

[106] S. T. Iqbal, X. S. Zheng, and B. P. Bailey. "Task-evoked Pupillary Response to Mental Workload in Human-Computer Interaction." In: *Extended Abstracts of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2004)*. ACM. 2004, pp. 1477–1480. DOI: 10.1145/985921.986094.

[107] ISO/IEC 25023:2016. *Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Measurement of system and software product quality*. (Last access: September 2019). ISO, Geneva, Switzerland. URL: https://www.iso.org/standard/35747.html.

[108] ISO/IEC/IEEE 29148:2018. *Systems and software engineering — Life cycle processes — Requirements engineering*. (Last access: September 2019). ISO, Geneva, Switzerland. URL: https://www.iso.org/standard/72089.html.

[109] iStarLab. (Last access: September 2019). URL: https://github.com/microlina/iStarLab.

[110] I. Jacobson. "Object-Oriented Software Engineering – A Use Case Driven Approach." In: *Proceedings of the 10th International Conference on Technology of Object-Oriented Languages and Systems (TOOLS 1993)*. 1993, p. 333.

[111] I. Jacobson, M. Christerson, P. Jonsson, and G. Övergaard. *Object-Oriented Software Engineering – A Use Case Driven Approach*. Addison-Wesley, 1992. ISBN: 978-0-201-54435-0.

[112] JASP. (Last access: September 2019). URL: https://jasp-stats.org.

[113] A. Jedlitschka, M. Ciolkowski, and D. Pfahl. "Reporting Experiments in Software Engineering." In: *Guide to Advanced Empirical Software Engineering*. Ed. by F. Shull, J. Singer, and D. I. K. Sjøberg. 1st ed. London, UK: Springer, 2008, pp. 201–228. ISBN: 978-1-84800-043-8. DOI: 10.1007/978-1-84800-044-5_8.

[114] W. Jernigan, A. Horvath, M. Lee, M. Burnett, T. Cuilty, S. Kuttal, A. Peters, I. Kwan, F. Bahmani, and A. Ko. "A Principled Evaluation for a Principled Idea Garden." In: *Proceedings of the IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC 2015)*. IEEE. 2015, pp. 235–243. DOI: 10.1109/VLHCC.2015.7357222.

[115] M. Jørgensen, T. Dybå, K. Liestøl, and D. I. K. Sjøberg. "Incorrect Results in Software Engineering Experiments: How to Improve Research Practices." In: *Journal of Systems and Software* 116 (2016), pp. 133–145. DOI: 10.1016/j.jss.2015.03.065.

[116] N. Juristo and A. M. Moreno. *Basics of Software Engineering Experimentation*. Springer Science & Business Media, 2013. ISBN: 978-1-4757-3304-4. DOI: 10.1007/978-1-4757-3304-4.

[117] S. H. Kan. *Metrics and Models in Software Quality Engineering*. 2nd ed. Addison-Wesley, 2003. ISBN: 978-0-201-63339-9.

[118] J. E. Kasser. "Object-Oriented Requirements Engineering and Management." In: *Proceedings of the Systems Engineering Test and Evaluation Conference*. 2003.

[119] C. Kelleher, R. Pausch, R. Pausch, and S. Kiesler. "Storytelling Alice Motivates Middle School Girls to Learn Computer Programming." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2007)*. ACM, 2007, pp. 1455–1464. ISBN: 978-1-59593-593-9. DOI: 10.1145/1240624.1240844.

[120] K. Kevic, B. M. Walters, T. R. Shaffer, B. Sharif, D. C. Shepherd, and T. Fritz. "Tracing Software Developers' Eyes and Interactions for Change Tasks." In: *Proceedings of the 10th Joint Meeting on Foundations of Software Engineering*. ACM. 2015, pp. 202–213. DOI: 10.1145/2786805.2786864.

[121] B. Kitchenham, L. Madeyski, D. Budgen, J. Keung, P. Brereton, S. M. Charters, S. Gibbs, and A. Pohthong. "Robust Statistical Methods for Empirical Software Engineering." In: *Empirical Software Engineering* 22.2 (2017), pp. 579–630. DOI: 10.1007/s10664-016-9437-5.

[122] B. Kitchenham, L. Madeyski, and P. Brereton. "Problems with Statistical Practice in Human-Centric Software Engineering Experiments." In: *Proceedings of the 23rd Evaluation and Assessment on Software Engineering (EASE 2019)*. ACM, 2019, pp. 134–143. DOI: 10.1145/3319008.3319009.

[123] B. A. Kitchenham and S. L. Pfleeger. "Personal Opinion Surveys." In: *Guide to Advanced Empirical Software Engineering*. Ed. by F. Shull, J. Singer, and D. I. K. Sjøberg. 1st ed. London, UK: Springer, 2008, pp. 63–92. ISBN: 978-1-84800-043-8. DOI: 10.1007/978-1-84800-044-5_3.

[124] J. Klingner. "Fixation-aligned Pupillary Response Averaging." In: *Proceedings of the ACM Symposium on Eye Tracking Research & Applications (ETRA 2010)*. ACM. 2010, pp. 275–282. DOI: 10.1145/1743666.1743732.

[125] A. J. Ko, S. Krishnamurhti, G. C. Murphy, and J. Siegmund. "Human-Centric Development of Software Tools." In: *Dagstuhl Reports* 5.5 (2016). DOI: 10.4230/DagRep.5.5.115.

[126] G. Kotonya and I. Sommerville. *Requirements Engineering: Processes and Techniques*. 1st ed. Wiley Publishing, 1998. ISBN: 978-0471972082.

[127] A. F. Kramer. "Physiological Metrics of Mental Workload: A Review of Recent Progress." In: *Multiple-Task Performance*. Ed. by D. L. Damos. 1st ed. Taylor & Francis, 1991, pp. 279–328. ISBN: 0-85066-757-7.

[128] J. Krogstie, G. Sindre, and H. Jørgensen. "Process Models Representing Knowledge for Action: A Revised Quality Framework." In: *European Journal of Information Systems* 15.1 (2006), pp. 91–102. DOI: 10.1057/palgrave.ejis.3000598.

[129] A. van Lamsweerde. "Goal-Oriented Requirements Engineering: A Guided Tour." In: *Proceedings of the 5th IEEE International Symposium on Requirements Engineering (ISRE 2001)*. IEEE, 2001, pp. 249–262. DOI: 10.1109/ISRE.2001.948567.

[130] A. van Lamsweerde. *Requirements Engineering: From System Goals to UML Models to Software Specifications*. 1st ed. Wiley Publishing, 2009. ISBN: 978-8126545896.

[131] C. F. J. Lange, M. A. M. Wijns, and M. R. V. Chaudron. "A Visualization Framework for Task-Oriented Modeling Using UML." In: *Proceedings of the 40th Annual Hawaii International Conference on System Sciences (HICSS 2007)*. IEEE. 2007, pp. 289–299. DOI: 10.1109/HICSS.2007.44.

[132] iStar Language. *i* vs iStar 2.0 at a glance*. (Last access: September 2019). URL: https://sites.google.com/site/istarlanguage/diff.

[133] M. Li and B.-L. Lu. "Emotion Classification Based on Gamma-band EEG." In: *Proceedings of the International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE. 2009, pp. 1223–1226. DOI: 10.1109/IEMBS.2009.5334139.

[134] O. I. Lindland, G. Sindre, and A. Solvberg. "Understanding Quality in Conceptual Modeling." In: *IEEE Software* 11.2 (1994), pp. 42–49. DOI: 10.1109/52.268955.

[135] L. Liu and E. Yu. "Designing Information Systems in Social Context: A Goal and Scenario Modelling Approach." In: *Information Systems* 29.2 (2004), pp. 187–203. DOI: 10.1016/S0306-4379(03)00052-8.

[136] J. Lockerbie, N. A. M. Maiden, J. Engmann, D. Randall, S. Jones, and D. Bush. "Exploring the Impact of Software Requirements on System-wide Goals: A Method Using Satisfaction Arguments and i* Goal Modelling." In: *Requirements Engineering* 17 (2012), pp. 227–254. DOI: 10.1007/s00766-011-0138-8.

[137] A. Luque-Casado, J. C. Perales, D. Cárdenas, and D. Sanabria. "Heart Rate Variability and Cognitive Processing: The Autonomic Response to Task Demands." In: *Biological Psychology* 113 (2016), pp. 83–90. DOI: 10.1016/j.biopsycho.2015.11.013.

[138] F. H. Martini and E. F. Bartholomew. *Essentials of Anatomy and Physiology*. 7th ed. Pearson, 2016. ISBN: 9781292156934.

[139] R. Matulevičius and P. Heymans. "Visually Effective Goal Models Using KAOS." In: *Proceedings of the 26th International Conference on Conceptual Modeling (ER 2007)*. Springer. 2007, pp. 265–275. DOI: 10.1007/978-3-540-76292-8_32.

[140] T. Mayerhofer. "Testing and Debugging UML Models based on fUML." In: *Proceedings of the 34th International Conference on Software Engineering (ICSE 2012)*. IEEE. 2012, pp. 1579–1582. DOI: 10.1109/ICSE.2012.6227032.

[141] D. McDuff, A. Karlson, A. Kapoor, A. Roseway, and M. Czerwinski. "AffectAura: An Intelligent System for Emotional Memory." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2012)*. ACM. 2012, pp. 849–858.

[142] P. Mohagheghi, V. Dehlen, and T. Neple. "Definitions and Approaches to Model Quality in Model-based Software Development – A Review of Literature." In: *Information and Software Technology* 51.12 (2009), pp. 1646–1669. ISSN: 0950-5849. DOI: 10.1016/j.infsof.2009.04.004.

[143] M. Monperrus, J. M. Jézéquel, B. Baudry, J. Champeau, and B. Hoeltzener. "Model-driven Generative Development of Measurement Software." In: *Software & Systems Modeling* 10.4 (2011), pp. 537–552. DOI: 10.1007/s10270-010-0165-9.

[144] D. L. Moody. "Theoretical and Practical Issues in Evaluating the Quality of Conceptual Models: Current State and Future Directions." In: *Data & Knowledge Engineering* 55.3 (2005), pp. 243–276. DOI: 10.1016/j.datak.2004.12.005.

[145] D. L. Moody. "The "Physics" of Notations: Toward a Scientific Basis for Constructing Visual Notations in Software Engineering." In: *IEEE Transactions on Software Engineering* 35.6 (2009), pp. 756–779. DOI: 10.1109/TSE.2009.67.

[146] K. Muldner, W. Burleson, and K. VanLehn. ""Yes!": Using Tutor and Sensor Data to Predict Moments of Delight During Instructional Activities." In: *Proceedinds of the 18th International Conference on User Modeling, Adaptation, and Personalization (UMAP 2010)*. Springer. 2010, pp. 159–170. DOI: 10.1007/978-3-642-13470-8_16.

[147] S. C. Müller and T. Fritz. "Stuck and Frustrated or in Flow and Happy: Sensing Developers' Emotions and Progress." In: *Proceedings of the 37th International Conference on Software Engineering (ICSE 2015)*. IEEE, 2015, pp. 688–699. DOI: 10.1109/ICSE.2015.334.

[148] S. C. Müller and T. Fritz. "Using (Bio) Metrics to Predict Code Quality Online." In: *Proceedings of the 38th International Conference on Software Engineering (ICSE 2016)*. ACM. 2016, pp. 452–463. DOI: 10.1145/2884781.2884803.

[149] M. Murugappan, M. Rizon, R. Nagarajan, S. Yaacob, I. Zunaidi, and D. Hazry. "EEG Feature Extraction for Classifying Emotions using FCM and FKM." In: *International Journal of Computers and Communications* 1.2 (2007), pp. 21–25.

[150] M. Murugappan, R. Nagarajan, and S. Yaacob. "Modified Energy Based Time-Frequency Features for Classifying Human Emotions Using EEG." In: *Proceedings of the International Conference on Man-Machine Systems (ICoMMS 2009)*. 2009, pp. 1–5.

[151] NaPiRE. *NaPiRE – Naming the Pain in Requirements Engineering*. (Last access: September 2019). URL: http://napire.org/.

[152] NaPiRE. *NaPiRE Data visualisation*. (Last access: September 2019). URL: http://napire.org/#/explore.

[153] NASA Task Load Index (TLX) Paper and Pencil Package. (Last access: September 2019). URL: https://humansystems.arc.nasa.gov/groups/TLX/downloads/TLX_pappen_manual.pdf.

[154] NASA-TLX in HTML and JavaScript. (Last access: September 2019). URL: https://www.keithv.com/software/nasatlx/.

[155] NASA-TLX iOS App. (Last access: September 2019). URL: https://humansystems.arc.nasa.gov/groups/TLX/tlxapp.php.

[156] NeuroSky MindWave EEG headset. (Last access: September 2019). URL: http://neurosky.com/biosensors/eeg-sensor/biosensors/.

[157] N. Nourbakhsh, Y. Wang, F. Chen, and R. A. Calvo. "Using Galvanic Skin Response for Cognitive Load Measurement in Arithmetic and Reading Tasks." In: *Proceedings of the 24th Australian Computer-Human Interaction Conference (OZCHI 2019)*. ACM. 2012, pp. 420–423. DOI: 10.1145/2414536.2414602.

[158] U. Obaidellah and M. A. Haek. "Evaluating Gender Difference on Algorithmic Problems Using Eye-tracker." In: *Proceedings of the ACM Symposium on Eye Tracking Research & Applications (ETRA 2018)*. ACM, 2018, pp. 1–8. DOI: 10.1145/3204493.3204537.

[159] Object Management Group. *Object Constraint Language (OCL)*. (Last access: September 2019). URL: http://www.omg.org/spec/OCL/.

[160] Object Management Group. *OMG Unified Modeling Language (OMG UML), Infrastructure, V2.1.2*. (Last access: September 2019). URL: http://www.omg.org/spec/UML/2.4.1/Infrastructure/PDF.

[161] OpenSignals. (Last access: September 2019). URL: http://bitalino.com/en/software.

[162] F. G. W. C. Paas and J. J. G. van Merriënboer. "The Efficiency of Instructional Conditions: An Approach to Combine Mental Effort and Performance Measures." In: *Human Factors: The Journal of the Human Factors and Ergonomics Society* 35.4 (1993), pp. 737–743. DOI: 10.1177/001872089303500412.

[163] F. G. W. C. Paas and J. J. G. van Merriënboer. "Instructional Control of Cognitive Load in the Training of Complex Cognitive Tasks." In: *Educational Psychology Review* 6.4 (1994), pp. 351–371. DOI: 10.1007/BF02213420.

[164] F. Pajares and M. D. Miller. "Role of Self-Efficacy and Self-Concept Beliefs in Mathematical Problem Solving: A Path Analysis." In: *Journal of Educational Psychology* 86.2 (1994), pp. 193–203. DOI: 10.1037/0022-0663.86.2.193.

[165] S. Papavlasopoulou, K. Sharma, and M. N. Giannakos. "Coding Activities for Children: Coupling Eye-tracking with Qualitative Data to Investigate Gender Differences." In: *Computers in Human Behavior* (2019). DOI: 10.1016/j.chb.2019.03.003.

[166] C. Parnin. "Subvocalization-toward Hearing the Inner Thoughts of Developers." In: *Proceedings of the 19th IEEE International Conference on Program Comprehension (ICPC 2011)*. IEEE. 2011, pp. 197–200. DOI: 10.1109/ICPC.2011.49.

[167] N. Paternoster, C. Giardino, M. Unterkalmsteiner, T. Gorschek, and P. Abrahamsson. "Software Development in Startup Companies: A Systematic Mapping Study." In: *Information & Software Technology* 56.10 (2014), pp. 1200–1218. DOI: 10.1016/j.infsof.2014.04.014.

[168] R. Petrusel and J. Mendling. "Eye-Tracking the Factors of Process Model Comprehension Tasks." In: *Proceedings of the 25th International Conference on Advanced Information Systems Engineering (CAiSE 2013)*. 2013, pp. 224–239. DOI: 10.1007/978-3-642-38709-8_15.

[169] J. Pimentel and J. Castro. "piStar Tool – A Pluggable Online Tool for Goal Modeling." In: *Proceedings of the IEEE International Requirements Engineering Conference (RE 2018)*. IEEE. 2018, pp. 498–499. DOI: 10.1109/RE.2018.00071.

[170] K. Pohl. *Requirements engineering: fundamentals, principles, and techniques*. 1st ed. Springer, 2010. ISBN: 978-3642125775.

[171] A. Poole and L. J. Ball. "Eye Tracking in HCI and Usability Research." In: *Encyclopedia of Human Computer Interaction* 1 (2006), pp. 211–219. DOI: 10.4018/978-1-59140-562-7.ch034.

[172] C. M. Privitera and L. W. Stark. "Algorithms for Defining Visual Regions-of-Interest: Comparison with Eye Fixations." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22.9 (2000), pp. 970–982. DOI: 10.1109/34.877520.

[173] PROMISE Software Engineering Repository. (Last access: September 2019). URL: http://promise.site.uottawa.ca/SERepository/.

[174] Python. (Last access: September 2019). URL: https://www.python.org.

[175] R. Radach, J. Hyona, and H. Deubel. *The Mind's Eye: Cognitive and Applied Aspects of Eye Movement Research*. 1st ed. Elsevier, 2003. ISBN: 978-0444510204.

[176] R. Ramos, J. Castro, J. Araújo, A. Moreira, F. Alencar, E. Santos, R. Penteado, S Carlos, and S Paulo. "AIRDoc – An Approach to Improve Requirements Documents." In: *22th Brazilian Symposium on Software Engineering*. 2008.

[177] A. Rashid, P. Sawyer, A. Moreira, and J. Araújo. "Early Aspects: A Model for Aspect-Oriented Requirements Engineering." In: *Proceedings of the IEEE Joint International Conference on Requirements Engineering*. IEEE. 2002, pp. 199–202. DOI: 10.1109/ICRE.2002.1048526.

[178] K. Rayner. "Eye Movements in Reading and Information Processing: 20 Years of Research." In: *Psychological Bulletin* 124.3 (1998), pp. 372–422. DOI: 10.1037/0033-2909.124.3.372.

[179] G. B. Reid and T. E. Nygren. "The Subjective Workload Assessment Technique: A Scaling Procedure for Measuring Mental Workload." In: *Advances in Psychology* 52 (1988), pp. 185–218. DOI: 10.1016/S0166-4115(08)62387-0.

[180] B. Reuderink, C. Mühl, and M. Poel. "Valence, Arousal and Dominance in the EEG During Game Play." In: *International Journal of Autonomous and Adaptive Communications Systems* 6.1 (2013), pp. 45–62. DOI: 10.1504/IJAACS.2013.050691.

[181] J. Robertson. "The Influence of a Game-making Project on Male and Female Learners' Attitudes to Computing." In: *Computer Science Education* 23.1 (2013), pp. 58–83. DOI: 10.1080/08993408.2013.774155.

[182] S. Rubio, E. Díaz, J. Martín, and J. M. Puente. "Evaluation of Subjective Mental Workload: A Comparison of SWAT, NASA-TLX, and Workload Profile Methods." In: *Applied Psychology* 53.1 (2004), pp. 61–86. DOI: 10.1111/j.1464-0597.2004.00161.x.

[183] P. Runeson, M. Höst, A. Rainer, and B. Regnell. *Case Study Research in Software Engineering: Guidelines and Examples*. 1st ed. Hoboken, NJ, USA: Wiley, 2012. ISBN: 978-1-118-10435-4.

[184] K. Ryu and R. Myung. "Evaluation of Mental Workload with a Combined Measure Based on Physiological Indices During a Dual Task of Tracking and Mental Asrithmetic." In: *International Journal of Industrial Ergonomics* 35.11 (2005), pp. 991–1009. DOI: 10.1016/j.ergon.2005.04.005.

[185] M. Santos, C. Gralha, M. Goulão, J. Araújo, A. Moreira, and J. Cambeiro. "What is the Impact of Bad Layout in the Understandability of Social Goal Models?" In: *Proceedings of the IEEE 24th International Requirements Engineering Conference (RE 2016)*. IEEE. 2016, pp. 206–215. DOI: 10.1109/RE.2016.51.

[186] M. Santos, C. Gralha, M. Goulão, and J. Araújo. "Increasing the Semantic Transparency of the KAOS Goal Model Concrete Syntax." In: *Proceedings of the 37th International Conference on Conceptual Modeling (ER 2018)*. Springer, 2018, pp. 424–439. DOI: 10.1007/978-3-030-00847-5_30.

[187] M. Santos, C. Gralha, M. Goulão, J. Araújo, and A. Moreira. "On the Impact of Semantic Transparency on Understanding and Reviewing Social Goal Models." In: *Proceedings of the 26th IEEE International Requirements Engineering Conference (RE 2018)*. IEEE, 2018, pp. 228–239. DOI: 10.1109/RE.2018.00031.

[188] A. G. Schissler, H. Nguyen, T. Nguyen, J. Petereit, and V. Gardeux. "Statistical Software." In: *Wiley StatsRef: Statistics Reference Online* (2019), pp. 1–11. DOI: 10.1002/9781118445112.stat00527.pub2.

[189] S. Schmidt. "Shall We Really Do It Again? The Powerful Concept of Replication is Neglected in the Social Sciences." In: *Review of General Psychology* 13.2 (2009), pp. 90–100. DOI: 10.1037/a0015108.

[190] S. Schmidt and H. Walach. "Electrodermal Activity (EDA): State-of-the-art Measurement and Techniques for Parapsychological Purposes." In: *Journal of Parapsychology* 64.2 (2000), p. 139.

[191] Z. Sharafi, Z. Soh, Y.-G. Guéhéneuc, and G. Antoniol. "Women and men – Different But Equal: On the Impact of Identifier Style on Source Code Reading." In: *Proceedings of the 20th IEEE International Conference on Program Comprehension (ICPC 2012)*. IEEE. 2012, pp. 27–36. DOI: 10.1109/ICPC.2012.6240505.

[192] Z. Sharafi, A. Marchetto, A. Susi, G. Antoniol, and Y.-G. Guéhéneuc. "An Empirical Study on the Efficiency of Graphical vs. Textual Representations in Requirements Comprehension." In: *Proceedings of the 21st International Conference on Program Comprehension (ICPC 2013)*. IEEE. 2013, pp. 33–42. DOI: 10.1109/ICPC.2013.6613831.

[193] Z. Sharafi, Z. Soh, and Y.-G. Guéhéneuc. "A Systematic Literature Review on the Usage of Eye-tracking in Software Engineering." In: *Information & Software Technology* 67 (2015), pp. 79–107. DOI: 10.1016/j.infsof.2015.06.008.

[194] Z. Sharafi, T. Shaffer, B. Sharif, et al. "Eye-Tracking Metrics in Software Engineering." In: *Proceedings of the Asia-Pacific Software Engineering Conference (APSEC 2015)*. IEEE. 2015, pp. 96–103. DOI: 10.1109/APSEC.2015.53.

[195] B. Sharif. "Empirical Assessment of UML Class Diagram Layouts Based on Architectural Importance." In: *Proceeding of the 27th International Conference on Software Maintenance (ICSM 2011)*. IEEE. 2011, pp. 544–549. DOI: 10.1109/ICSM.2011.6080828.

[196] B. Sharif and J. Maletic. "An Eye Tracking Study on camelCase and under_score Identifier Styles." In: *Proceedings of the IEEE 18th International Conference on Program Comprehension (ICPC 2010)*. IEEE. 2010, pp. 196–205. DOI: 10.1109/ICPC.2010.41.

[197] B. Sharif and J. Maletic. "An Eye Tracking Study on the Effects of Layout in Understanding the Role of Design Patterns." In: *Proceedings of the 26th IEEE International Conference on Software Maintenance (ICSM 2010)*. IEEE. 2010, pp. 1–10. DOI: 10.1109/ICSM.2010.5609582.

[198] B. Sharif and J. Maletic. "The Effects of Layout on Detecting the Role of Design Patterns." In: *Proceedings 23rd IEEE Conference on Software Engineering Education and Training (CSEE&T)*. IEEE. 2010, pp. 41–48. DOI: 10.1109/CSEET.2010.23.

[199] B. Sharif and J. I. Maletic. "An Empirical Study on the Comprehension of Stereotyped UML Class Diagram Layouts." In: *Proceedings of the 17th IEEE International Conference on Program Comprehension (ICPC 2009)*. IEEE. 2009, pp. 268–272. DOI: 10.1109/ICPC.2009.5090055.

[200] B. Sharif, M. Falcone, and J. I. Maletic. "An Eye-tracking Study on the Role of Scan Time in Finding Source Code Defects." In: *Proceedings of the ACM Symposium on Eye Tracking Research & Applications (ETRA 2012)*. ACM. 2012, pp. 381–384. DOI: 10.1145/2168556.2168642.

[201] Y. Shi, N. Ruiz, R. Taib, E. Choi, and F. Chen. "Galvanic Skin Response (GSR) as an Index of Cognitive Load." In: *Extended Abstracts of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2007)*. ACM. 2007, pp. 2651–2656. DOI: 10.1145/1240866.1241057.

[202] D. Showkat and C. Grimm. "Identifying Gender Differences in Information Processing Style, Self-efficacy, and Tinkering for Robot Tele-operation." In: *Proceedings of the 15th International Conference on Ubiquitous Robots (UR 2018)*. IEEE. 2018, pp. 443–448. DOI: 10.1109/URAI.2018.8441766.

[203] F. Shull, V. Basili, J. Carver, J. C. Maldonado, G. H. Travassos, M. Mendonça, and S. Fabbri. "Replicating Software Engineering Experiments: Addressing the Tacit Knowledge Problem." In: *Proceedings of the International Symposium on Empirical Software Engineering (ISESE 2002)*. IEEE. 2002, pp. 7–16. DOI: 10.1109/ISESE.2002.1166920.

[204] J. Siegmund, C. Kästner, S. Apel, C. Parnin, A. Bethmann, T. Leich, G. Saake, and A. Brechmann. "Understanding Understanding Source Code with Functional Magnetic Resonance Imaging." In: *Proceedings of the 36th International Conference on Software Engineering (CAiSE 2014)*. ACM. 2014, pp. 378–389. DOI: 10.1145/2568225.2568252.

[205] F. Q. B. da Silva, M. Suassuna, A. C. C. França, A. M. Grubb, T. B. Gouveia, C. V. Monteiro, and I. E. dos Santos. "Replication of Empirical Studies in Software Engineering Research: A Systematic Mapping Study." In: *Empirical Software Engineering* 19.3 (2014), pp. 501–557. DOI: 10.1007/s10664-012-9227-7.

[206] L. F. da Silva, A. Moreira, J. Araújo, C. Gralha, M. Goulão, and V. Amaral. "Exploring Views for Goal-Oriented Requirements Comprehension." In: *Proceedings of the 35th International Conference on Conceptual Modeling (ER 2016)*. 2016, pp. 149–163. DOI: 10.1007/978-3-319-46397-1_12.

[207] D. I. K. Sjøberg, J. E. Hannay, O. Hansen, V. B. Kampenes, A. Karahasanovic, N.-K. Liborg, and A. C. Rekdal. "A Survey of Controlled Experiments in Software Engineering." In: *IEEE Transactions on Software Engineering* 31.9 (2005), pp. 733–753. DOI: 10.1109/TSE.2005.97.

[208] R. P. Sloan, P. A. Shapiro, E. Bagiella, S. M. Boni, M. Paik, J. T. Bigger Jr, R. C. Steinman, and J. M. Gorman. "Effect of Mental Stress Throughout the Day on Cardiac Autonomic Control." In: *Biological Psychology* 37.2 (1994), pp. 89–99. DOI: 10.1016/0301-0511(94)90024-8.

[209] B. de Smet, L. Lempereur, Z. Sharafi, Y.-G. Guéhéneuc, G. Antoniol, and N. Habra. "Taupe: Visualizing and Analyzing Eye-tracking Data." In: *Science of Computer Programming* 79 (2014), pp. 260–278. DOI: 10.1016/j.scico.2012.01.004.

[210] M. E. Smith and A. Gevins. "Neurophysiologic Monitoring of Mental Workload and Fatigue During Operation of a Flight Simulator." In: *Proceedings of the Biomonitoring for Physiological and Cognitive Performance during Military Operations*. Vol. 5797. International Society for Optics and Photonics. 2005, pp. 116–127. DOI: 10.1117/12.602181.

[211] I. Sommerville. *Software Engineering*. 10th ed. Pearson, 2015. ISBN: 9780133943030.

[212] I. Sommerville and P. Sawyer. "Viewpoints: Principles, Problems and a Practical Approach to Requirements Engineering." In: *Annals of Software Engineering* 3 (1997), pp. 101–130. DOI: 10.1023/A:1018946223345.

[213] Statistics for the evaluation of requirements modelss. (Last access: September 2019). URL: http://microlina.github.io/statistics/home.html.

[214] H. Störrle. "On the Impact of Layout Quality to Understanding UML Diagrams." In: *Proceedings of the IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC 2011)*. IEEE. 2011, pp. 135–142. DOI: 10.1109/VLHCC.2011.6070390.

[215] H. Störrle. "On the Impact of Layout Quality to Understanding UML Diagrams: Diagram Type and Expertise." In: *Proceedings of the IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC 2012)*. IEEE. 2012, pp. 49–56. DOI: 10.1109/VLHCC.2012.6344480.

[216] H. Störrle. "On the Impact of Layout Quality to Understanding UML Diagrams: Size Matters." In: *Proceedings of the International Conference on Model Driven Engineering Languages and Systems (MoDELS 2014)*. Springer. 2014, pp. 518–534. DOI: 10.1007/978-3-319-11653-2_32.

[217] H. Störrle, N. Baltsen, H. Christoffersen, and A. Maier. "On the Impact of Diagram Layout: How Are Models Actually Read?" In: *Proceedings of the International Conference on Model Driven Engineering Languages and Systems (MoDELS 2014)*. 2014, pp. 31–35.

[218] R. Subramanyam and M. S. Krishnan. "Empirical Analysis of CK Metrics for Object-Oriented Design Complexity: Implications for Software Defects." In: *IEEE Transactions on Software Engineering* 29.4 (2003), pp. 297–310. DOI: 10.1109/TSE.2003.1191795.

[219] D. Szafir and B. Mutlu. "Pay Attention! Designing Adaptive Agents that Monitor and Improve User Engagement." In: *Proceedings of the SIGCHI Conference on Human factors in Computing Systems (CHI 2012)*. ACM. 2012, pp. 11–20. DOI: 10.1145/2207676.2207679.

[220] D. S. Tan, M. Czerwinski, and G. Robertson. "Women Go With the (Optical) Flow." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2013)*. ACM. 2003, pp. 209–215. DOI: 10.1145/642611.642649.

[221] W. O. Tatum. *Handbook of EEG Interpretation*. 2nd ed. Demos Medical Publishing, 2014. ISBN: 9781620700167.

[222] The Eye Tribe. (Last access: September 2019). URL: https://theeyetribe.com/.

[223] The Eye Tribe. *Eye Tracking 101*. (Last access: September 2019). URL: http://theeyetribe.com/dev.theeyetribe.com/dev.theeyetribe.com/general/index.html.

[224] The R project for statistical computing. (Last access: September 2019). URL: https://www.r-project.org.

[225] W. F. Tichy. "Should Computer Scientists Experiment More?" In: *IEEE Computer* 31.5 (1998), pp. 32–40. DOI: 10.1109/2.675631.

[226] S. Tiwari and A. Gupta. "A Systematic Literature Review of Use Case Specifications Research." In: *Information & Software Technology* 67 (2015), pp. 128–158. DOI: 10.1016/j.infsof.2015.06.004.

[227] TLX@NASA. (Last access: September 2019). URL: https://humansystems.arc.nasa.gov/groups/TLX/tlxpaperpencil.php.

[228] K. Torii, K.-i. Matsumoto, K. Nakakoji, Y. Takada, S. Takada, and K. Shima. "Ginger2: An Environment for Computer-aided Empirical Software Engineering." In: *IEEE Transactions on Software Engineering* 25.4 (1999), pp. 474–492. DOI: 10.1109/32.799942.

[229] G. Torkzadeh and X. Koufteros. "Factorial Validity of a Computer Self-Efficacy Scale and the Impact of Computer Training." In: *Educational and Psychological Measurement* 54.3 (1994), pp. 813–821. DOI: 10.1177/0013164494054003028.

[230] G. H. Travassos and M. O. Barros. "Contributions of In Virtuo and In Silico Experiments for the Future of Empirical Studies in Software Engineering." In: *Proceedings of the 2nd Workshop on Empirical Software Engineering the Future of Empirical Studies in Software Engineering*. 2003, pp. 117–130.

[231] S. Uchida, H. Kudo, and A. Monden. "An Experiment and an Analysis of Debugging Process with Periodic Interviews." In: *Proceedings of Software Symposium*. Vol. 98. 1998, pp. 53–58.

[232] Usability Evaluation Framework. (Last access: September 2019). URL: https://microlina.github.io/Framework.

[233] H. Uwano, M. Nakamura, A. Monden, and K.-i. Matsumoto. "Analyzing Individual Performance of Source Code Review Using Reviewers' Eye Movement." In: *Proceedings of the ACM Symposium on Eye Tracking Research & Applications (ETRA 2006)*. ACM. 2006, pp. 133–140. DOI: 10.1145/1117309.1117357.

[234] S. Vegas, C. Apa, and N. Juristo. "Crossover Designs in Software Engineering Experiments: Benefits and Perils." In: *IEEE Transactions on Software Engineering* 42.2 (2016), pp. 120–135. DOI: 10.1109/TSE.2015.2467378.

[235] J. A. Veltman and A. W. K. Gaillard. "Physiological Workload Reactions to Increasing Levels of Task Difficulty." In: *Ergonomics* 41.5 (1998), pp. 656–669. DOI: 10.1080/001401398186829.

[236] M. Vorvoreanu, L. Zhang, Y.-H. Huang, C. Hilderbrand, Z. Steine-Hanson, and M. Burnett. "From Gender Biases to Gender-Inclusive Design: An Empirical Investigation." In: *Proceedings of the Conference on Human Factors in Computer Systems (CHI 2019)*. ACM, 2019, pp. 53–66. DOI: 10.1145/3290605.3300283.

[237] B. Walters, T. Shaffer, B. Sharif, and H. Kagdi. "Capturing Software Traceability Links from Developers' Eye Gazes." In: *Proceedings of the 22nd International Conference on Program Comprehension (ICPC 2014)*. ACM. 2014, pp. 201–204. DOI: 10.1145/2597008.2597795.

[238] M.-T. Wang and J. L. Degol. "Gender Gap in Science, Technology, Engineering, and Mathematics (STEM): Current Knowledge, Implications for Practice, Policy, and Future Directions." In: *Educational psychology review* 29.1 (2017), pp. 119–140. DOI: 10.1007/s10648-015-9355-x.

[239] J. Warmer and A. Kleppe. *The Object Constraint Language: Precise Modeling with UML*. 1st ed. Addison-Wesley, 1999. ISBN: 978-0-201-37940-2.

[240] A. T. Welford. "Mental Workload as a Function of Demand, Capacity, Strategy and Skill." In: *Ergonomics* 21.3 (1978), pp. 151–167. DOI: 10.1080/00140137808931710.

[241] i* wiki. *Events*. (Last access: September 2019). URL: http://istarwiki.org/tiki-index.php?page=Events.

[242] i* wiki. *i* Guide*. (Last access: September 2019). URL: http://istar.rwth-aachen.de/tiki-index.php?page=iStarQuickGuide.

[243] i* wiki. *Who is Who*. (Last access: September 2019). URL: http://istarwiki.org/tiki-index.php?page=Who+is+Who.

[244] G. F. Wilson. "Applied Use of Cardiac and Respiration Measures: Practical Considerations and Precautions." In: *Biological Psychology* 34.2-3 (1992), pp. 163–178. DOI: 10.1016/0301-0511(92)90014-L.

[245] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén. *Experimentation in Software Engineering*. 2nd ed. London, UK: Springer, 2012. ISBN: 978-3-642-29044-2.

[246] Y. Y. Yeh and C. D. Wickens. "Dissociation of Performance and Subjective Measures of Workload." In: *Human Factors: The Journal of the Human Factors and Ergonomics Society* 30.1 (1988), pp. 111–120. DOI: 10.1177/001872088803000110.

[247] E. Yu. "Modelling Strategic Relationships for Process Reengineering." Doctoral dissertation. Canada: Department of Computer Science, University of Toronto, 1995.

[248] E. Yu. "Towards Modelling and Reasoning Support for Early-Phase Requirements Engineering." In: *Proceedings of the 3rd IEEE International Symposium on Requirements Engineering (ISRE 1997)*. IEEE, 1997, pp. 226–235. DOI: 10.1109/ISRE.1997.566873.

[249] E. Yu. "Agent Orientation as a Modelling Paradigm." In: *Wirtschaftsinformatik* 43.2 (2001), pp. 123–132. DOI: 10.1007/BF03250789.

[250] S. Yusuf, H. Kagdi, and J. I. Maletic. "Assessing the Comprehension of UML Class Diagrams Via Eye Tracking." In: *Proceeding of the 15th IEEE International Conference on Program Comprehension (ICPC 2007)*. IEEE. 2007, pp. 113–122. DOI: 10.1109/ICPC.2007.10.

[251] Zenodo – Empirical Software Engineering. (Last access: September 2019). URL: https://zenodo.org/communities/empirical-software-engineering/.

APPENDIX A

Auxiliary metrics for iStar 2.0 Models Complexity and Completeness Evaluation

In this Appendix, we provide the auxiliary metrics required for both the correct calculation of the metrics, as well as to compute other auxiliary metrics presented in Chapter 4. Some of the auxiliary metrics are related with more than one question, being each metric presented individually in the order in which it is referred in the text in the Subsection 4.2.2. In cases where auxiliary metrics require other metrics, they will be defined shortly thereafter, when they have not been previously defined. Each metric has a name, an informal definition in natural language, and a formal definition in OCL.

Table A.1: Auxiliary metric NEOAB

Metrics	**NEOAB** – ***N**umber of **E**lements **O**utside **A**ctors' **B**oundaries*
Informal definition	Total number of elements outside an actor's boundary in the SD/SR model

Formal definition	`context ISTAR`
	`def:NEOAB():Integer = self.hasNode ->`
	`select(n:Node | n.oclIsKindOf(Element)) -> size()`

Table A.2: Auxiliary metric NEIAB

Metric	**NEIAB** – ***N**umber of **E**lements **I**nside **A**ctors' **B**oundaries*
Informal definition	Total number of elements inside an actor's boundary in the SD/SR model

Formal definition	`context ISTAR`
	`def:NEIAB():Integer = self.hasNode ->`
	`select(n:Node | n.oclIsKindOf(Actor)) ->`
	`iterate(n:Node; total:Integer = 0 |`
	`let nea:Integer = n.oclAsType(Actor).NEA() in`
	`total + nea)`

Requires	**NEA** – **N**umber of **E**lements of an **A**ctor

Table A.3: Auxiliary metric NGWDI

Metric	**NGWDI** – ***N**umber of **G**oals **W**ith **D**ecompositions **I**nside*
Informal definition	Number of goals with decompositions (refinement links) inside an actor's boundary in the SR model

Formal definition	`context Actor`
	`def:NGWDI():Integer = self.hasElement ->`
	` select(e:Element | e.oclIsKindOf(Goal) and`
	`  e.oclAsType(Goal).NDG() > 0) -> size()`

Requires	**NDG** – ***N**umber of **D**ecompositions of a **G**oal*

Table A.4: Auxiliary metric NQWDI

Metric	**NQWDI** – ***N**umber of **Q**ualities **W**ith **D**ecompositions **I**nside*
Informal definition	Number of qualities with decompositions inside an actor's boundary in the SR model

Formal definition	`context Actor`
	`def:NQWDI():Integer = self.hasElement ->`
	` select(e:Element | e.oclIsKindOf(Quality) and`
	`  e.oclAsType(Quality).NDQ() > 0) -> size()`

Requires	**NDQ** – **N**umber of **D**ecompositions of a **Q**uality

Table A.5: Auxiliary metric NTWDI

Metric	**NTWDI** – ***N**umber of **T**asks **W**ith **D**ecompositions **I**nside*
Informal definition	Number of task with decompositions inside an actor's boundary in the SR model

Formal definition	`context Actor`
	`def:NTWDI():Integer = self.hasElement ->`
	` select(e:Element | e.oclIsKindOf(Task) and`
	`  e.oclAsType(Task).NDT() > 0) -> size()`

Requires	**NDT** – **N**umber of **D**ecompositions of a **T**ask

Table A.6: Auxiliary metric NOD

Metric	**NOD** – ***N**umber of **O**utgoing **D**ependencies*
Informal definition	Number outgoing dependencies of an actor in the SD/SR model
Formal definition	`context Actor` `def:NOD():Integer = self.NODAI() + self.NODEI()`
Requires	**NODAI** – **N**umber of **O**utgoing **D**ependencies of an **A**ctor **I**tself (AM A.7) **NODEI** – **N**umber of **O**utgoing **D**ependencies of an **E**lement **I**nside (AM A.8)

Table A.7: Auxiliary metric NODAI

Metric	**NODAI** – *Number of Outgoing Dependencies of an Actor Itself*
Informal definition	Number of outgoing dependencies of an actor itself in the SD/SR model

Formal definition	`context Actor`
	`def:NODAI():Integer = self.actorDependency ->`
	`select(dl:DependencyLink |`
	`dl.oclIsKindOf(DependerLink)) -> size()`

Table A.8: Auxiliary metric NODEI

Metric	**NODEI** – *Number of Outgoing Dependencies of an Element Inside*
Informal definition	Number of outgoing dependencies of an element inside an actor's boundary in the SD/SR model

Formal definition	`context Actor`
	`def:NODEI():Integer = self.hasElement ->`
	`select(e:Element | e.oclIsKindOf(Element)) ->`
	`iterate(e:Element; total:Integer = 0 |`
	`let nodepe:Integer = e.oclAsType(Element).NODepE() in`
	`total + nodepe)`

Requires	**NODepE** – **N**umber of **O**utgoing **Dep**endencies of an **E**lement (AM A.9)

Table A.9: Auxiliary metric NODepE

Metric	**NODepE** – *Number of Outgoing Dependencies of an Element*
Informal definition	Total number of outgoing dependencies of an element inside an actor's boundary in the SD/SR model
Formal definition	`context Element` `def:NODepE():Integer = self.NODE() + self.NODEA()`
Requires	**NODE** – **N**umber of **O**utgoing **D**ependencies from an **E**lement (AM A.10) **NODEA** – **N**umber of **O**utgoing **D**ependencies from an **E**lement to an **A**ctor (AM A.11)

Table A.10: Auxiliary metric NODE

Metric	**NODE** – *Number of Outgoing Dependencies from an Element*
Informal definition	Number of outgoing dependencies from element to an element inside an actor's boundary in the SD/SR model

Formal definition	`context Element`
	`def:NODE():Integer = self.elementDependency ->`
	`select(dl:DependencyLink |`
	`dl.oclIsKindOf(DepElemLink)) -> size()`

Table A.11: Auxiliary metric NODEA

Metric	**NODEA** – ***N**umber of **O**utgoing **D**ependencies from an **E**lement to an **A**ctor*
Informal definition	Number of outgoing dependencies from element to an actor in the SD/SR model

Formal definition	`context Element`
	`def:NODEA():Integer = self.elementDependency ->`
	`select(dl:DependencyLink |`
	`dl.oclIsKindOf(DependeeLink)) -> size()`

Table A.12: Auxiliary metric ND

Metric	**ND** – ***N**umber of **D**ependencies*
Informal definition	Number of dependencies of an actor in the SD/SR model
Formal definition	`context Actor` `def:ND():Integer = self.NID() + self.NOD()`
Requires	**NID** – **N**umber of **I**ncoming **D**ependencies (AM A.13) **NOD** – **N**umber of **O**utgoing **D**ependencies (AM A.6)

Table A.13: Auxiliary metric NID

Metric	**NID** - ***N**umber of **I**ncoming **D**ependencies*
Informal definition	Number of incoming dependencies of an actor in the SD/SR model
Formal definition	`context Actor` `def:NID():Integer = self.NIDAI() + self.NIDEI()`
Requires	**NIDAI** – **N**umber of **I**ncoming **D**ependencies of an **A**ctor **I**tself (AM A.14) **NIDEI** – **N**umber of **I**ncoming **D**ependencies of an **E**lement **I**nside (AM A.15)

Table A.14: Auxiliary metric NIDAI

Metric	**NIDAI** – ***N**umber of **I**ncoming **D**ependencies of an **A**ctor **I**tself*
Informal definition	Number of incoming dependencies of an actor itself in the SD/SR model

Formal definition	`context Actor`
	`def:NIDAI():Integer = self.actorDependency ->`
	`select(dl:DependencyLink |`
	`dl.oclIsKindOf(DependeeLink)) -> size()`

Table A.15: Auxiliary metric NIDEI

Metric	**NIDEI** – ***N**umber of **I**ncoming **D**ependencies of an **E**lement **I**nside*
Informal definition	Number of incoming dependencies of an element inside an actor's boundary in the SD/SR model

Formal definition	`context Actor`
	`def:NIDEI():Integer = self.hasElement ->`
	`select(e:Element | e.oclIsKindOf(Element)) ->`
	`iterate(e:Element; total:Integer = 0 |`
	`let nidepe:Integer = e.oclAsType(Element).NIDepE() in`
	`total + nidepe)`

Requires	**NIDepE** – **N**umber of **I**ncoming **Dep**endencies of an **E**lement (AM A.16)

Table A.16: Auxiliary metric NIDepE

Metric	**NIDepE** – ***N**umber of **I**ncoming **Dep**endencies of an **E**lement*
Informal definition	Total number of incoming dependencies of an element inside an actor's boundary in the SD/SR model
Formal definition	`context Element` `def:NIDepE():Integer = self.NIDE() + self.NIDEA()`
Requires	**NIDE** – **N**umber of **I**ncoming **D**ependencies of an **E**lement (AM A.17) **NIDEA** – **N**umber of **I**ncoming **D**ependencies to an **E**lement from an **A**ctor (AM A.18)

Table A.17: Auxiliary metric NIDE

Metric	**NIDE** – ***N**umber of **I**ncoming **D**ependencies of an **E**lement*
Informal definition	Total number of incoming dependencies from an element to an element inside an actor's boundary in the SD/SR model

Formal definition	`context Element`
	`def:NIDE():Integer = self.secondElementDependency ->`
	` select(dl:DependencyLink |`
	`  dl.oclIsKindOf(DepElemLink)) -> size()`

Table A.18: Auxiliary metric NIDEA

Metric	**NIDE** – ***N**umber of **I**ncoming **D**ependencies of an **E**lement*
Informal definition	Total number of incoming dependencies from an element to an actor in the SD/SR model

Formal definition	`context Element`
	`def:NIDEA():Integer = self.elementDependency ->`
	` select(dl:DependencyLink |`
	`  dl.oclIsKindOf(DependerLink)) -> size()`

Table A.19: Auxiliary metric NEIAgB

Metric	**NEIAgB** – ***N**umber of **E**lements **I**nside **Ag**ents' **B**oundaries*
Informal definition	Total number of elements inside agents' boundaries

Formal definition	`context ISTAR`
	`def:NEIAgB():Integer = self.hasNode ->`
	` select(n:Node | n.oclIsKindOf(Agent)) ->`
	`  iterate(n:Node; total:Integer = 0 |`
	`  let neia:Integer = n.oclAsType(Agent).NEIA() in total+neia)`

Requires	**NEIAg** – **N**umber of **E**lements **I**nside an **A**gent

Table A.20: Auxiliary metric NEIRB

Metric	**NEIRB** – ***N**umber of **E**lements **I**nside **R**oles' **B**oundaries*
Informal definition	Total number of elements inside roles' boundaries

Formal definition	`context ISTAR`
	`def:NEIRB():Integer = self.hasNode ->`
	`select(n:Node | n.oclIsKindOf(Role)) ->`
	`iterate(n:Node; total : Integer = 0 |`
	`let neir : Integer = n.oclAsType(Role).NEIR() in total+neir)`

Requires	**NEIR** – **N**umber of **E**lements **I**nside a **R**ole

Table A.21: Auxiliary metric NAgents

Metric	**NAgents** – ***N**umber of **Agents***
Informal definition	Number of agents in the SD/SR model

Formal definition	`context ISTAR`
	`def:NAgents():Integer = self.hasNode ->`
	`select(n:Node | n.oclIsKindOf(Agent)) -> size()`

Table A.22: Auxiliary metric NRoles

Metric	**NRoles** – ***N**umber of **Roles***
Informal definition	Number of roles in the SD/SR model

Formal definition	`context ISTAR`
	`def:NRoles():Integer = self.hasNode ->`
	`select(n:Node | n.oclIsKindOf(Role)) -> size()`

Table A.23: Auxiliary metric NGWD

Metric	**NGWD** – ***N**umber of **G**oals **W**ith **D**ecompositions*
Informal definition	Total number of goals with decompositions (refinement links) in the SR model

Formal definition	`context ISTAR`
	`def:NGWD():Integer = self.hasNode ->`
	`select(n:Node | n.oclIsKindOf(Actor)) ->`
	`iterate(n:Node; total:Integer = 0 |`
	`let ngwdi:Integer = n.oclAsType(Actor).NGWDI() in`
	`total + ngwdi)`

Requires	**NGWDI** – **N**umber of **G**oals **W**ith **D**ecompositions **I**nside (AM A.3)

Table A.24: Auxiliary metric NGWQ

Metric	**NGWQ** – ***N**umber of **G**oals **W**ith **Q**ualifications*
Informal definition	Total number of goals with qualifications in the SR model

Formal definition	`context ISTAR`
	`def:NGWQ():Integer = self.hasNode ->`
	`select(n:Node | n.oclIsKindOf(Actor)) ->`
	`iterate(n:Node; total:Integer = 0 |`
	`let ngwqi:Integer = n.oclAsType(Actor).NGWQI() in`
	`total + ngwqi)`

Requires	**NGWQI** – **N**umber of **G**oals **W**ith **Q**ualifications **I**nside (AM A.25)

Table A.25: Auxiliary metric NGWQI

Metric	**NGWQ** – ***N**umber of **G**oals **W**ith **Q**ualifications* **I**nside
Informal definition	Number of goals with qualifications inside an actor's boundary in the SR model

Formal definition	`context Actor`
	`def:NGWQI():Integer = self.hasElement ->`
	`select(e:Element | e.oclIsKindOf(Goal) and`
	`e.oclAsType(Goal).NQG() > 0) -> size()`

Requires	**NQG** – **N**umber of **Q**ualifications of a **G**oal (AM A.26)

Table A.26: Auxiliary metric NQG

Metric	**NQG** – ***N**umber of **Q**ualifications of a **G**oal*
Informal definition	Number of qualifications associated with a goal in the SR model

Formal definition	`context Goal`
	`def:NQG():Integer = self.elementQualification ->`
	`select(re:Qualification | re.oclIsKindOf(Qualification)) -> size`
	`↪ ()`

Table A.27: Auxiliary metric NGIAB

Metric	**NGIAB** – ***N**umber of **G**oals **I**nside **A**ctors' **B**oundaries*
Informal definition	Total of goals inside actors' boundaries in the SR model

Formal definition	`context ISTAR`
	`def:NGIAB():Integer = self.hasNode ->`
	`select(n:Node | n.oclIsKindOf(Actor)) ->`
	`iterate(n:Node; total:Integer = 0 |`
	`let ngi:Integer = n.oclAsType(Actor).NGI() in total + ngi)`

Requires	**NGI** – ***N**umber of **G**oals **I**nside* (AM A.28)

Table A.28: Auxiliary metric NGI

Metric	**NGI** – ***N**umber of **G**oals **I**nside*
Informal definition	Number of goals inside an actor's boundaries in the SR model

Formal definition	`context Actor`
	`def:NGI():Integer = self.hasElement ->`
	`select(e:Element | e.oclIsKindOf(Goal)) -> size()`

Table A.29: Auxiliary metric NQWD

Metric	**NQWD** – ***N**umber of **Q**ualities **W**ith **D**ecompositions*
Informal definition	Total number of qualities with decompositions in the SR model

Formal definition	`context ISTAR`
	`def:NQWD():Integer = self.hasNode ->`
	`select(n:Node | n.oclIsKindOf(Actor)) ->`
	`iterate(n:Node; total:Integer = 0 |`
	`let nswdi:Integer = n.oclAsType(Actor).NSWDI() in`
	`total + nswdi)`

Requires	**NQWDI** – ***N**umber of **Q**ualities **W**ith **D**ecompositions **I**nside* (MA A.4)

Table A.30: Auxiliary metric NQWQ

Metric	**NQWQ** – *Number of Qualities With Qualifications*
Informal definition	Total number of qualities with qualifications in the SR model

Formal definition	`context ISTAR`
	`def:NQWQ():Integer = self.hasNode ->`
	` select(n:Node | n.oclIsKindOf(Actor)) ->`
	`  iterate(n:Node; total:Integer = 0 |`
	`   let nqwqi:Integer = n.oclAsType(Actor).NQWQI() in`
	`    total + nqwqi)`

Requires	**NQWQI** – **N**umber of **Q**ualities **W**ith **Q**ualifications **I**nside (AM A.31)

Table A.31: Auxiliary metric NQWQI

Metric	**NQWQI** – *Number of Qualities With Qualifications* **I**nside
Informal definition	Number of qualities with qualifications inside an actor's boundary in the SR model

Formal definition	`context Actor`
	`def:NQWQI():Integer = self.hasElement ->`
	` select(e:Element | e.oclIsKindOf(Goal) and`
	`  e.oclAsType(Goal).NQG() > 0) -> size()`

Requires	**NQG** – **N**umber of **Q**ualifications of a **Q**quality (AM A.32)

Table A.32: Auxiliary metric NQQ

Metric	**NQQ** – *Number of Qualifications of a Quality*
Informal definition	Number of qualifications associated with a quality in the SR model

Formal definition	`context Quality`
	`def:NQQ():Integer = self.qualitificationQuality ->`
	` select(re:Qualification | re.oclIsKindOf(Qualification)) ->`
	`  size ()`

Table A.33: Auxiliary metric NQIAB

Metric	**NQIAB** – **N**umber of **Q**ualities **I**nside **A**ctors' **B**oundaries
Informal definition	Total number of qualities inside actors' boundaries in the SR model

Formal definition	`context ISTAR`
	`def:NQIAB():Integer = self.hasNode ->`
	` select(n:Node | n.oclIsKindOf(Actor)) ->`
	`  iterate(n:Node; total:Integer = 0 |`
	`   let nqi : Integer = n.oclAsType(Actor).NQI() in total + nqi)`

Requires	**NQI** – **N**umber of **Q**ualities **I**nside (AM A.34)

Table A.34: Auxiliary metric NQI

Metric	**NQI** – *Number of Qualities Inside*
Informal definition	Number of qualities inside an actor's boundaries in the SR model

Formal definition	`context Actor`
	`def:NQI():Integer = self.hasElement ->`
	` select(e:Element | e.oclIsKindOf(Quality)) -> size()`

Table A.35: Auxiliary metric NAWEI

Metric	**NAWEI** – ***N**umber of **A**ctors **W**ith **E**lements **I**nside*
Informal definition	Total number of actors with elements inside its boundaries in the SR model

Formal definition	`context ISTAR`
	`def:NAWEI():Integer = self.hasNode ->`
	` select(n:Node | n.oclIsKindOf(Actor) and`
	`  n.oclAsType(Actor).NEI() > 0) -> size()`

Requires	**NEI** – **N**umber of **E**lements **I**nside (AM A.36)

Table A.36: Auxiliary metric NEI

Metric	**NEI** – ***N**umber of **E**lements **I**nside*
Informal definition	Number of elements inside an actor's boundary in the SR model

Formal definition	`context Actor`
	`def:NEI():Integer = self.hasElement ->`
	` select(e:Element | e.oclIsKindOf(Element)) -> size()`

Table A.37: Auxiliary metric PAWUEI

Metric	**PAWUEI** – ***P**ercentage of **A**ctors **W**ith **U**nconnected **E**lements **I**nside*
Informal definition	Percentage of actors with unconnected elements inside its boundaries in the SR model

Formal definition	`context ISTAR::PAWUEI`
	`pre: self.NAct() > 0`
	
	`context ISTAR`
	`def:PAWUEI():Double = self.NAWUEI() / self.NAWEI()`

Requires	**NAWUEI** – ***N**umber of **A**ctors **W**ith **U**nconnected **E**lements **I**nside* (AM A.38) **NAWEI** – ***N**umber of **A**ctors **W**ith **E**lements **I**nside* (AM A.35)

Table A.38: Auxiliary metric NAWUEI

Metric	**NAWUEI** – ***N**umber of **A**ctors **W**ith **U**nconnected **E**lements **I**nside*
Informal definition	Number of actors with unconnected elements inside its boundaries in the SR model

Formal definition	`context ISTAR`
	`def:NAWUEI():Integer = self.hasNode ->`
	` select(n:Node | n.oclIsKindOf(Actor) and`
	`  n.oclAsType(Actor).NUEI() > 0) -> size()`

Requires	**NUEI** – **N**umber of **U**nconnected **E**lements **I**nside (MA A.39)

Table A.39: Auxiliary metric NUEI

Metric	**NUEI** – ***N**umber of **U**nconnected **E**lements **I**nside*
Informal definition	Number of unconnected elements inside an actor's boundary in the SR model

Formal definition	`context Actor`
	`def:NUEI():Integer = self.NUGI() + self.NUQI() +`
	`self.NUTI() + self.NURI()`

Requires	**NUGI** – **N**umber of **U**nconnected **G**oals **I**nside (AM A.40) **NUQI** – **N**umber of **U**nconnected **Q**ualities **I**nside (AM A.42) **NUTI** – **N**umber of **U**nconnected **T**asks **I**nside (AM A.44) **NURI** – **N**umber of **U**nconnected **R**esources **I**nside (MA A.45)

Table A.40: Auxiliary metric NUGI

Metric	**NUGI** – ***N**umber of **U**nconnected **G**oals **I**nside*
Informal definition	Number of unconnected goals inside an actor's boundary in the SR model

Formal definition	`context Actor`
	`def:NUGI():Integer = self.hasElement ->`
	`select(e:Element | e.oclIsKindOf(Goal) and`
	`e.oclAsType(Goal).NLG() = 0) -> size()`

Requires	**NLG** – **N**umber of **L**inks of a **G**oal (AM A.41)

Table A.41: Auxiliary metric NLG

Metric	**NLG** – **N**umber of **L**inks of a **G**oal
Informal definition	Number of links of a goal in the SR model
Formal definition	`context Goal` `def:NLG():Integer = self.NDG() + self.NQG()`
Requires	**NDG** – **N**umber of **D**ecompositions of a **G**oal **NQG** – **N**umber of **Q**ualifications of a **G**oal (AM A.26)

Table A.42: Auxiliary metric NUQI

Metric	**NUSI** – ***N**umber of **U**nconnected **Q**ualities **I**nside*
Informal definition	Number of unconnected qualities inside an actor's boundary in the SR model

Formal definition	`context Actor`
	`def:NUQI():Integer = self.hasElement ->`
	`select(e:Element | e.oclIsKindOf(Quality) and`
	`e.oclAsType(Softgoal).NLQ() = 0) -> size()`

Requires	**NLQ** – **N**umber of **L**inks of a **Q**uality (AM A.43)

Table A.43: Auxiliary metric NLQ

Metric	**NLQ** – ***N**umber of **L**inks of a **Q**uality*
Informal definition	Number of links of a quality in the SR model
Formal definition	`context Quality` `def:NLQ():Integer = self.NDQ() + self.NQQ()`
Requires	**NDQ** – **N**umber of **D**ecompositions of a **Q**uality **NQQ** – **N**umber of **Q**ualifications of a **Q**uality

Table A.44: Auxiliary metric NUTI

Metric	**NUTI** – ***N**umber of **U**nconnected **T**asks **I**nside*
Informal definition	Number of unconnected tasks inside an actor's boundary in the SR model

Formal definition	`context Actor`
	`def:NUTI():Integer = self.hasElement ->`
	` select(e:Element | e.oclIsKindOf(Task) and`
	` e.oclAsType(Task).NDT() = 0) -> size()`

Requires	**NDT** – **N**umber of **D**ecompositions of a **T**ask

Table A.45: Auxiliary metric NURI

Metric	**NURI** – ***N**umber of **U**nconnected **R**esources **I**nside*
Informal definition	Number of unconnected resources inside an actor's boundary in the SR model

Formal definition	`context Actor`
	`def:NURI():Integer = self.hasElement ->`
	`select(e:Element | e.oclIsKindOf(Resource) and`
	`e.oclAsType(Resource).NDR() = 0) -> size()`

Requires	**NDR** – **N**umber of **D**ecompositions of a **R**esource

Table A.46: Auxiliary metric NDR

Metric	**NDR** – ***N**umber of **D**ecompositions of a **R**esource*
Informal definition	Number of decompositions associated with a resource in the SR model

Formal definition	`context Resource`
	`def:NDG():Integer = self.neededByResource ->`
	` select(re:NeededBy | re.oclIsKindOf(NeededBy)) -> size ()`

Table A.47: Auxiliary metric NAWDOA

Metric	**NAWDOA** – ***N**umber of **A**ctors **W**ith Dependencies **O**r **A**ssociations*
Informal definition	Total number of actors with dependencies or association links in the SD/SR model

Formal definition	`context ISTAR`
	`def:NAWDOA():Integer = self.hasNode ->`
	` select(n:Node | n.oclIsKindOf(Actor) and`
	`  (n.oclAsType(Actor).ND() > 0 or`
	`  n.oclAsType(Actor).NA() > 0)) -> size()`

Requires	**ND** – **N**umber of **D**ependencies (AM A.12) **NA** – **N**umber of **A**ssociations (AM A.48)

Table A.48: Auxiliary metric NA

Metric	**NA** – ***N**umber of **A**ssociations*
Informal definition	Number of associations of an actor in the SD/SR model
Formal definition	`context Actor` `def:NA():Integer = self.NISA() + self.NPIn()`
Requires	**NISA** – **N**umber of **ISA** (AM A.49) **NPIn** – **N**umber of **P**articates **In** (AM A.50)

Table A.49: Auxiliary metric NISA

Metric	**NISA** – **N**umber of **ISA**
Informal definition	Number of Is-A associations of an actor in the SD/SR model

Formal definition	`context Actor`
	`def:NISA():Integer = self.actorISA ->`
	`select(isa:ISA | isa.oclIsKindOf(ISA)) -> size()`

Table A.50: Auxiliary metric NPIn

Metric	**NPIn** – **N**umber of **P**articipates **In**
Informal definition	Number of Participate-in associations of an actor in the SD/SR model

Formal definition	`context Actor`
	`def:NPIn():Integer = self.actorParticipatesIn ->`
	`select(pi:ParticipatesIn | pi.oclIsKindOf(ParticipatesIn)) ->`
	`size()`